

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMA

ESTRATÉGIAS DE DETECÇÃO E DIAGNÓSTICO
DE FALHAS EM SISTEMAS DINÂMICOS

Este exemplar corresponde a redação final da tese defendida por <u>Walmir Matos Caminhas</u> e aprovada pela Comissão Julgada em <u>24.1.11.1997</u> . <u>Hermano M. F. Tavares</u> Orientador
--

Autor: Walmir Matos Caminhas

Orientador: Prof. Dr. Hermano M. F. Tavares

Co-Orientador: Prof. Dr. Fernando Antônio Campos Gomide

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Campinas como requisito parcial para obtenção de Título de Doutor em Engenharia Elétrica.

Novembro, 1997



7155-086

DE	BC
RAMADA:	UNICAMP
	C. 146e
EA	
BO/	329/98
	395/98
	D
	X
	R\$ 11,00
	01/03/98
PD	

CM-00106584-B

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C146e

Caminhas, Walmir Matos

Estratégias de detecção e diagnóstico de falhas em sistemas dinâmicos / Walmir Matos Caminhas.--Campinas, SP: [s.n.], 1997.

Orientadores: Hermano M. F. Tavares, Fernando Antônio Campos Gomide.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Falha de sistema (Engenharia)* 2. Sistemas dinâmicos diferenciais* 3. Redes neurais (Computação)* 4. Conjuntos nebulosos* I. Tavares, Hermano M. F. II. Gomide, Fernando Antônio Campos. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

AGRADECIMENTOS

Aos Professores Hermano Tavares e Fernando Gomide pela orientação, amizade e oportunidade de trabalharmos juntos.

Ao Professor Fernando Von Zuben pelas críticas e sugestões.

Aos ex-mestrandos do CPGEE/UFMG Marcus Felipe Pereira e Herlon Ayres Carmargo pelo fornecimento de dados e resultados experimentais da planta piloto do Sistema de Tanques Iterativos.

Ao colega Ricardo H. C. Takahashi pelas valiosas discussões e sugestões.

Aos amigos do Laboratório de Alta Tensão da UNICAMP Ângelo, Marcus Vinícius, Francisco Brito, Eliane e Pissolato e aos seus familiares pela amizade e calorosa acolhida durante estes quatro anos.

Ao Departamento de Engenharia Elétrica da UFMG pela liberação integral para realização do Programa de Doutorado.

Ao PICD/UFMG/CAPES pelo auxílio financeiro.

À Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo auxílio financeiro para participação de eventos, que foram muito importante para o desenvolvimento do trabalho.

À minha mulher Josely
aos meus filhos Daniel e Larissa
aos meus pais Joaquim e Ildete.

RESUMO

A detecção e diagnóstico de falhas (DDF) em sistemas dinâmicos baseados em redundância analítica são tratadas sob diversas abordagens, fundamentadas em conceitos básicos que incluem: espaço de paridade; estimação de estados; estimação de parâmetros; sistemas especialistas; reconhecimento de padrões e outros. Neste trabalho a DDF é abordada sob a ótica de classificação de padrões. O problema de classificação de padrões pode ser resolvido utilizando-se diversas técnicas, tais como algoritmos heurísticos, técnicas de probabilidade, redes neurais, lógica *fuzzy* e outras. Aqui, para classificação de padrões, é proposta uma estrutura de rede *neurofuzzy and/or*, além de uma modificação na estrutura clássica da rede neural auto-organizada. São propostas três estratégias de sistema de detecção e diagnóstico de falhas. Na primeira, os padrões são formados a partir das informações de entrada e saída da planta. A segunda e a terceira são baseadas em sistema de classificação de padrões e geração residual (estratégias híbridas). Na segunda, os resíduos são gerados a partir de comparações entre os estados para diversos observadores em modos deslizantes. Na terceira, são utilizados parâmetros do modelo da planta, que são os pesos de uma rede neurofuzzy. As estruturas foram testadas em três sistemas dinâmicos, a saber: sistema de acionamento elétrico utilizando máquina de corrente contínua; sistema de acionamento elétrico utilizando máquina de corrente alternada e sistema de tanques interativos (acoplados). Os modelos matemáticos destes sistemas são não-lineares e de ordem 3, 7 e 2, respectivamente. São apresentados resultados de simulações de falhas para os três sistemas dinâmicos. Para o sistema de tanques interativos são apresentados resultados experimentais de detecção *off-line*. Os resultados de simulação e experimentais confirmam a eficiência das estratégias propostas e as credenciam como uma abordagem promissora na solução de problemas complexos de DDF.

ABSTRACT

Analytical redundancy for fault detection and diagnosis of dynamic systems, FDD, has been approached by several methodologies, state estimation, parameter estimation, expert systems, and pattern classification and recognition being typical examples. In particular, pattern classification and recognition methods adopt probabilistic, heuristic, neural, and fuzzy set based techniques as a solution framework. This work introduces the FDD as a pattern classification and recognition problem. Two neural approaches for pattern classification and recognition are introduced and compared: a neurofuzzy network composed by logical *and* and *or* neurons with a competitive learning, and a variation of a supervised Learning Vector Quantization (LVQ) network with N_1 , N_2 and N_∞ norms and a pruning scheme, respectively. In addition, three schemes are proposed to solve FDD problems. The first scheme uses input and output signals to classify the operational condition of the dynamic system. In this case mathematical models are not needed to detect and to diagnose faults. The second uses residues generated by comparisons among the states of several sliding mode observers. The third uses the parameters of a model of the system. The model parameters are the weights of a neurofuzzy network. The last two schemes constitute hybrid strategies to solve FDD problems. The schemes developed herein were tested in three different dynamic systems: a DC motor drive; an AC motor drive, and an interactive tank system. Simulation results are reported for these three example problems. Experimental results for off-line fault detection and diagnosis of the interactive tank are also included. The simulation and experimental results obtained confirmed the efficiency of the schemes developed, and their relevance as a promising approach to solve complex FDD problems as well.

SUMÁRIO

CAPÍTULO 1: DESCRIÇÃO DO TRABALHO

1.1- INTRODUÇÃO	...01
1.2- METODOLOGIA PROPOSTA	...02
1.3- ESTRUTURAÇÃO DO TEXTO	...10

CAPÍTULO 2: CLASSIFICAÇÃO DE PADRÕES USANDO LÓGICA FUZZY E REDE NEURAL

2.1 - INTRODUÇÃO	...12
2.2 - CLASSIFICAÇÃO DE PADRÕES USANDO LÓGICA FUZZY	...12
2.2.1 - Sistema <i>Fuzzy</i>	...12
2.2.2 - Classificação de Padrões Usando Sistema <i>Fuzzy</i>	...15
2.3 -IMPLEMENTAÇÃO DE SISTEMA DE CLASSIFICAÇÃO FUZZY VIA REDE NEUROFUZZY	...19
2.3.1 - Neurônios Lógicos	...20
2.3.2 - Rede <i>Neurofuzzy and/or</i>	...21
2.4 - APRENDIZADO DA REDE NEUROFUZZY AND/OR	...23
2.4.1 - Geração das Funções de Pertinência	...23
2.4.2 - Geração das Conexões da Rede e Inicialização dos Pesos	...27
2.4.3 - Determinação dos Neurônios <i>and/or</i> Ativos	...28
2.4.4 - <i>Fuzzificação</i>	...29
2.4.5 - Determinação da Classe Vencedora	...29
2.4.6 - Atualização dos Pesos	...30
2.5 - CLASSIFICAÇÃO DE PADRÕES USANDO REDE NEURAL	...31
2.5.1 - Estrutura da Rede Neural	...31
2.5.2 - Algoritmo de Aprendizado da Rede Neural Auto-Organizada	...32

2.6 - EXEMPLOS DIDÁTICOS DE CLASSIFICAÇÃO DE PADRÕES	...33
2.6.1 - Resultados para a Rede <i>Neurofuzzy and/or</i>	...33
2.6.2 - Resultados para a Rede Neural	...39
2.7 - ANÁLISE DE COMPLEXIDADE DAS ESTRUTURAS DE REDE	...42
2.8 - CONCLUSÃO	...43
 CAPÍTULO 3 : MODELAGEM DE FALHAS	
3.1 - INTRODUÇÃO	...44
3.2 - MODELAGEM DE FALHAS PARA O SISTEMA DE ACIONAMENTO C.C.	...44
3.2.1 - MODELO DO SISTEMA DE ACIONAMENTO C.C.	...44
3.2.2 - Modelagem de Falhas para o Sistema de Acionamento C.C.	...48
3.2.3 - Modelo Geral das Falhas no Sistema de Acionamento C.C.	...50
3.2.4 - Resultados de Simulações de Falhas no Sistema de Acionamento C.C.	...52
3.3 - MODELAGEM DE FALHAS PARA O SISTEMA DE ACIONAMENTO C.A.	...55
3.3.1 - Modelagem do Sistema de Acionamento C.A.	...55
3.3.2 - Modelagem de Falhas para o Sistema de Acionamento C.A.	...57
3.3.3 - Modelo Geral de Falhas para o Sistema de Acionamento C.A.	...61
3.3.4 - Resultados de Simulações de Falhas no Sistema de Acionamento C.A.	...62
3.4 - MODELAGEM DE FALHAS PARA O SISTEMA DE TANQUES INTERATIVOS	...65
3.4.1 - Descrição do Sistema	...65
3.4.2 - Modelo do Sistema de Tanques Interativos	...66
3.4.3 - Modelagem de Falhas para o Sistema de Tanques Interativos	...67
3.4.4 - Modelo Geral de Falhas para o Sistema de Tanques Interativos	...71
3.4.5 - Resultados de Simulações de Falhas no Sistema de Tanques Interativos	...72
3.5 - CONCLUSÃO	...75

**CAPÍTULO 4 : OBSERVADORES DE ESTADOS PARA SISTEMAS
INCERTOS: ABORDAGEM BASEADA NA TEORIA
DE MODOS DESLIZANTES**

4.1 - INTRODUÇÃO	...76
4.2 - OBSERVADORES EM MODOS DESLIZANTES PARA SISTEMAS DISCRETOS NÃO LINEARES INCERTOS	...76
4.3 - PROJETO DOS OBSERVADORES EM MODOS DESLIZANTES PARA O SISTEMA DE ACIONAMENTO C.C.	...79
4.3.1- Observador para Rejeição de Perturbações na Direção de x_1	...80
4.3.2 - Observador para Rejeição de Perturbações na Direção de x_2	...82
4.3.3 - Observador para Rejeição de Perturbações na Direção de x_3	...83
4.3.4 - Observador para Rejeição de Perturbações na Direção de x_1 e x_2	...83
4.3.5 - Observador para Rejeição de Perturbações na Direção de x_1 e x_3	...84
4.3.6 - Observador para Rejeição de Perturbações na Direção de x_2 e x_3	...85
4.4 - RESULTADOS DE SIMULAÇÕES	...85
4.5 - CONCLUSÃO	...88

**CAPÍTULO 5: IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS USANDO
NEO FUZZY NEURON**

5.1 - INTRODUÇÃO	...90
5.2 - <i>NEO FUZZY NEURON</i> (NFN)	...90
5.2.1 - Motivação	...90
5.2.2 - Estrutura do NFN	...91
5.2.3 - Aprendizado do NFN	...93
5.3 - <i>NEO FUZZY NEURON - GENERALIZADO</i> (NFNG)	...99
5.3.1 - Estrutura do NFNG	...99
5.3.2 - Treinamento do NFNG	...100
5.3.3 - Geração das Funções de Pertinência do NFNG	...101

5.4 - IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS	...102
5.4.1 - Estratégia de Treinamento do NFNG para Identificação de Sistemas Dinâmicos	...103
5.4.2 - Aplicação do NFNG em Identificação de Sistemas Dinâmicos	...103
5.5 - ANÁLISE DA COMPLEXIDADE DOS ALGORITMOS: NFNG; REDE NEURAL E REDE <i>NEUROFUZZY</i> DO WANG (1994)	...116
5.6 - CONCLUSÃO	...118
CAPÍTULO 6: DETECÇÃO E DIAGNÓSTICO DE FALHAS EM SISTEMAS DINÂMICOS	
6.1 - INTRODUÇÃO	...119
6.2 - ESTRUTURA GERAL DO SISTEMA DE DETECÇÃO E DIAGNÓSTICO DE FALHAS EM SISTEMAS DINÂMICOS	...119
6.3 - ESTRATÉGIA BASEADA EM ENTRADAS E SAÍDAS DA PLANTA E SISTEMA DE CLASSIFICAÇÃO DE PADRÕES	...120
6.3.1 - Detecção e Diagnóstico de Falhas no Sistema de Acionamento C.A	...119
6.3.2 - Detecção e Diagnóstico de Falhas no Sistema de Tanques Interativos	...126
6.4 - ESTRATÉGIA BASEADA EM OBSERVADORES DE ESTADOS EM MODOS DESLIZANTES E CLASSIFICAÇÃO DE PADRÕES	...132
6.5 - ESTRATÉGIA BASEADA EM ESTIMAÇÃO DE PARÂMETROS E CLASSIFICAÇÃO DE PADRÕES	...139
6.6 - CONCLUSÃO	...144
CAPÍTULO 7: CONCLUSÕES E CONSIDERAÇÕES FINAIS146
REFERÊNCIAS BIBLIOGRÁFICAS149
APÊNDICE A: PARÂMETROS DOS SISTEMAS DINÂMICOS157
APÊNDICE B: PARÂMETROS DO NFNG PARA O EXEMPLO 1160

CAPÍTULO 1

DESCRIÇÃO DO TRABALHO

1.1 - INTRODUÇÃO

Devido ao grande desenvolvimento tecnológico, principalmente da microeletrônica, os sistemas automatizados estão se tornando cada vez mais complexos e os algoritmos de controle cada vez mais sofisticados. O desenvolvimento de sistemas tolerantes a falhas é uma preocupação crescente, principalmente porque, além de fornecer uma maior segurança aos operadores ou usuários das plantas, proporcionam economia no que diz respeito a paradas para manutenção, reduzindo assim o preço final do produto. Neste sentido, um assunto que vem ganhando destaque nas duas últimas décadas é o desenvolvimento de técnicas para detecção e diagnóstico de falhas em sistemas dinâmicos (Frank, 1990). Uma área onde são encontradas muitas aplicações dessas técnicas é a aviação. Isto se justifica pelo fato de haver riscos de vida humana (Patton *et alii*, 1989).

A detecção de falhas em sistemas dinâmicos é tradicionalmente obtida por redundância de *hardware*, como, por exemplo, utilizando-se dois ou mais sensores para medir uma mesma grandeza. A partir da comparação dos valores medidos e utilizando uma lógica adequada é possível identificar a ocorrência de falha. As principais desvantagens dessa técnica são o aumento do custo e um acréscimo considerável no peso total da planta (Patton *et alii*, 1989). Esta última é um fator importante quando se trata de aplicação em missões espaciais. Um outro modo de detectar falhas é a redundância por *software* ou redundância analítica. Neste caso, no lugar de vários sensores são empregados, por exemplo, vários observadores para estimar a mesma grandeza. A partir da comparação dos valores estimados e medidos é possível a detecção, localização e diagnóstico de falhas (Patton *et alii*, 1989; Frank, 1990). Com relação a custos, este sistema é privilegiado já que normalmente utiliza recursos computacionais já presentes nas plantas modernas para sistemas supervisórios e algoritmos de controle (Patton *et alii*, 1989; Pereira, 1996). São várias as abordagens encontradas na literatura para detecção e diagnóstico de falhas em sistemas dinâmicos. Uma visão geral de algumas dessas abordagens pode ser encontrada em Frank (1990); Frank e Seliger (1991); Isermann (1984); Patton *et alii* (1989); Willsky (1976) e Himmelblau (1978), cuja leitura é recomendada.

Basicamente as estratégias de detecção de falhas em sistemas dinâmicos, que utilizam o conceito de redundância analítica, podem ser agrupadas em duas classes, baseadas em: modelos quantitativos do processo; modelos qualitativos (baseados no conhecimento) (Frank, 1990). Na prática os métodos baseados em modelos qualitativos, normalmente, são combinados com os métodos baseados em modelos quantitativos (Frank, 1990). Para a primeira classe, pode-se citar o emprego de: observadores de estados tipo Luenberger dedicados ou filtro de Kalman (Clark, 1978^{a,b}; Willsky, 1976; Frank e Keller, 1980; Mehra e Peshon, 1971; Pereira, 1996; Aguirre *et alii*, 1996); observadores robustos para entradas desconhecidas (Frank e Wünnenberg, 1989, Ge e Fang, 1988); estimação de parâmetros (Iserman, 1984; Camargo, 1997); técnicas de aprendizado (redes neurais e *neurofuzzy*) (Chow e Mangum, 1991; Goode e Chow, 1994; Bernieri *et alii*, 1995; Polycarpou e Vemuri, 1995;

Ulieru e Mrsic-Flogel, 1995; Pistauer, 1995; Tyan *et alii*, 1995; Liu e Yan, 1997) e outros. As técnicas baseadas em modelos qualitativos utilizam lógica clássica (Milne, 1987; Gonzalez *et alii*, 1986; Tzafestas, 1989; Vinson e Ungar, 1995; Betta *et alii*, 1995) ou lógica *fuzzy* (Vescovi e Robles, 1992; Umano *et alii*, 1994; Weber, 1994; Wu e Wu, 1994; Ulieru, 1994; David, 1994; Jonathan e Ungar, 1995).

O objetivo do trabalho é estudar e comparar algumas estratégias de detecção e diagnóstico de falhas em sistemas dinâmicos. As abordagens de detecção de falhas propostas são baseadas em conceitos básicos de: sistema de classificação de padrões; observador de estados e identificação sistemas dinâmicos.

1.2 - METODOLOGIA PROPOSTA

1.2.1 - ESTRUTURA BÁSICA

Na figura 1.1 é mostrado o diagrama de blocos da abordagem empregada para detecção e diagnóstico de falhas em sistemas dinâmicos, proposta neste trabalho. A idéia básica é tratar o problema de detecção de falhas como um problema de classificação de padrões. Os vetores u (entradas), y (saídas da planta) e y^* (saídas de referência) são utilizados para gerar o padrão, que é entrada do sistema de classificação. Serão abordadas duas estruturas de sistema de classificação de padrões: a primeira baseada em sistema *fuzzy*, implementado por uma rede *neurofuzzy and/or* (Caminhas *et alii*, 1995^{a,b}; Caminhas *et alii*, 1996^c), e a segunda baseada em uma rede neural auto-organizada com algoritmo de treinamento tipo LVQ ("Learning Vector Quantization") (Kohonen, 1990). Como as saídas do sistema de classificação de padrões são números, é utilizada uma tabela para as correspondentes conversões em expressões lingüísticas. Estas expressões lingüísticas são do tipo: se a saída do sistema de classificação é " i ", então a falha é no atuador " j ". Na figura 1.1. o vetor f é usado somente durante a fase de treinamento do sistema de classificação.

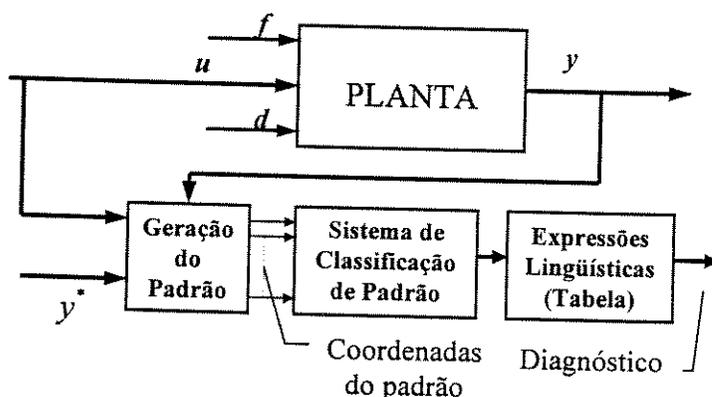


Fig. 1.1 - Estrutura do sistema de detecção e diagnóstico de falhas

A idéia básica desta abordagem pode ser compreendida com auxílio da figura 1.2. Para permitir uma visualização em duas dimensões, considere um sistema dinâmico ao qual pode-se associar um padrão de duas coordenadas para obter informação do seu funcionamento.

Considere as coordenadas $x_{p1}(t)$ e $x_{p2}(t)$, que podem ser entradas, saídas, estados estimados, parâmetros ou combinação destes. Estas coordenadas definem o espaço caracterizado pelo par ordenado $(x_{p1}(t), x_{p2}(t))$. Para este sistema, considere três situações de operação (classes): normal; falha tipo 1 e falha tipo 2. Estas situações são separadas pelas fronteiras evidenciadas na figura 1.2. Considere que a fronteira de classificação, obtida por um algoritmo qualquer de classificação de padrões, figura 1.2, é usada para detectar e isolar as falhas. Com base nas fronteiras de separação e de classificação, pode-se ilustrar os índices que permitem avaliar a qualidade de um sistema de detecção de falhas, a saber: *alarme falso*; *falha detectada e isolada corretamente*; *detecção da falha mas isolamento incorreto*; *não detecção da falha* e *tempo de detecção da falha*. Para analisar graficamente estas situações suponha que a situação inicial é operação normal do sistema em $t=t_0$, caracterizada pelo par $(x_{p1}(t_0), x_{p2}(t_0))$ representado pelo ponto P_1 . Considere as seguintes situações:

- operação do sistema no ponto P_2 : o ponto P_2 pertence à região, definida pela fronteira de separação, correspondente à operação com falha tipo 1. A classificação deste ponto baseada na fronteira de classificação também é de operação com falha tipo 1. Portanto, para a situação caracterizada pelo ponto P_2 a falha é detectada e isolada corretamente;
- o ponto P_3 , baseado na fronteira de classificação, é classificado como falha tipo 2, mas corresponde ao funcionamento em condição normal, caracterizando um *alarme falso*;
- o ponto P_5 é classificado como operação falha tipo 2. Como este ponto corresponde a situação de operação com falha tipo 1, neste caso a falha é detectada mas isolada incorretamente;
- o ponto P_4 representa uma *não detecção da falha*;
- o tempo de detecção da falha: se ocorrer uma falha tipo 1, representada pelo ponto P_2 , o tempo gasto para detectar a falha é definido pelo tempo que o sistema dinâmico levou de P_1 até atingir a fronteira de classificação.

Observa-se que, quanto mais próximo a fronteira de classificação estiver da fronteira de separação real, melhor será o sistema de detecção.

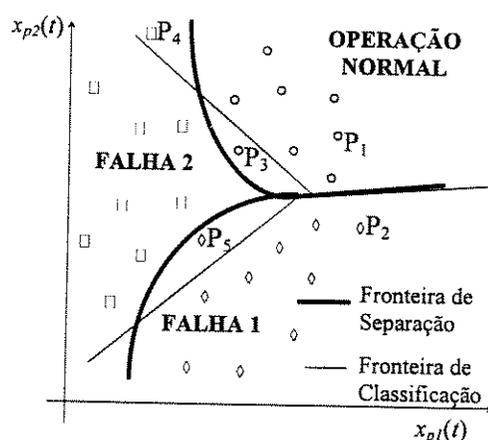


Fig. 1.2 - Ilustração de Classificação de Falhas

Para exemplificar considere o sistema dinâmico representado na figura 1.3. O sistema é composto por uma fonte de tensão e uma carga elétrica (resistor, R , e um indutor, L). Considere três situações possíveis: operação normal (N), curto-circuito nos terminais da fonte (F1) e rompimento do condutor que liga a fonte à carga (F2). Sem considerar o transitório de energização do circuito, para uma fonte de tensão:

$$v(t) = V_m \cdot \text{sen}(\omega \cdot t)$$

a corrente é dada por

$$i(t) = I_m \cdot \text{sen}(\omega \cdot t - \varphi), \text{ onde } \varphi = \tan^{-1}\left(\frac{\omega L}{R}\right)$$

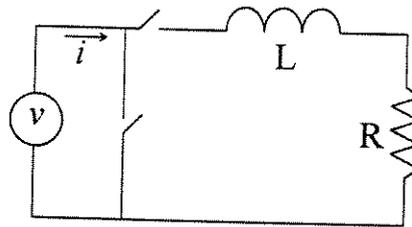


Fig. 1.3 - Sistema dinâmico ilustrativo

Na figura 1.4 é mostrado o gráfico de $v(t) \times i(t)$, valores normalizados. Verifica-se que a relação entre as duas grandezas é uma elipse, na condição de operação normal. A condição de curto-circuito em regime permanente é caracterizada pela reta $v(t)=0$. Já a condição de desligamento da carga é caracterizada pela reta $i(t)=0$. Se as grandezas são amostradas a relação $v(kT) \times i(kT)$ é dada pelos pontos mostrados na figura 1.5, onde T é o período de amostragem. Se, ao invés de se trabalhar com valores instantâneos, optar-se por valores eficazes, a relação $V_{\text{ef}}(kT) \times I_{\text{ef}}(kT)$ é mostrada no gráfico da figura 1.6. A separação dos padrões para esta última representação é mais fácil.

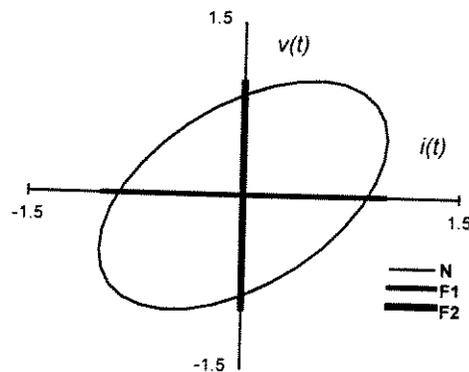


Fig. 1.4 - Curva $v(t) \times i(t)$ para o exemplo ilustrativo - Valores instantâneos contínuos

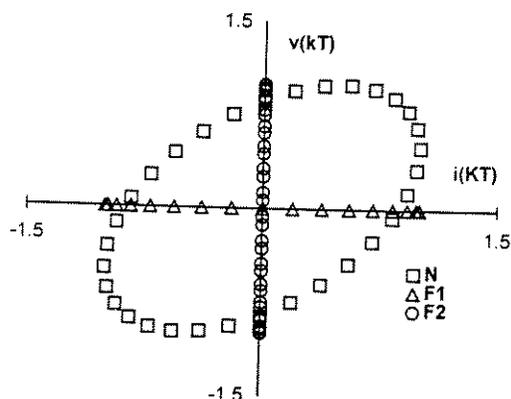


Fig. 1.5 - Curva $v(kT) \times i(kT)$ para o exemplo ilustrativo
Valores instantâneos amostrados

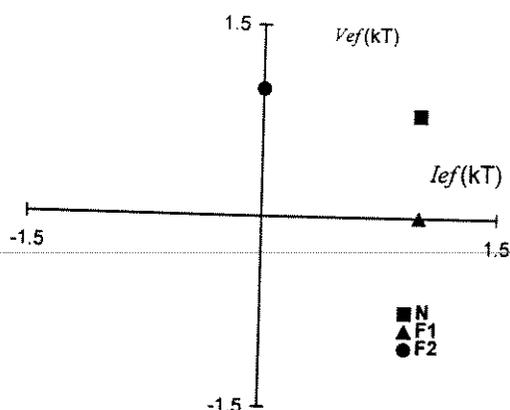


Fig. 1.6 - Curva $V_{ef}(kT) \times I_{ef}(kT)$ para o exemplo ilustrativo
Valores eficazes amostrados

Para este exemplo as falhas podem ser separadas individualmente. Pode haver situações, como a mostrada na figura 1.7, onde é mais fácil determinar os conjuntos de falhas que possuem um certo grau de similaridade (*cluster*) e posteriormente tentar separar as falhas individualmente. Para isto, é proposta uma estrutura distribuída, figura 1.8, composta de dois estágios: o primeiro faz o agrupamento das falhas que possuem um certo grau de afinidade (*clusterização*) e o segundo faz a classificação da falha, quando for possível. A idéia é dividir para conquistar, pois o emprego de somente um sistema de classificação poderia exigir um número muito grande de neurônios. A saída do primeiro estágio determina qual sistema de classificação do segundo estágio será utilizado. As entradas de um sistema de classificação “*j*” não são necessariamente as mesmas coordenadas do sistema de agrupamento de falhas.

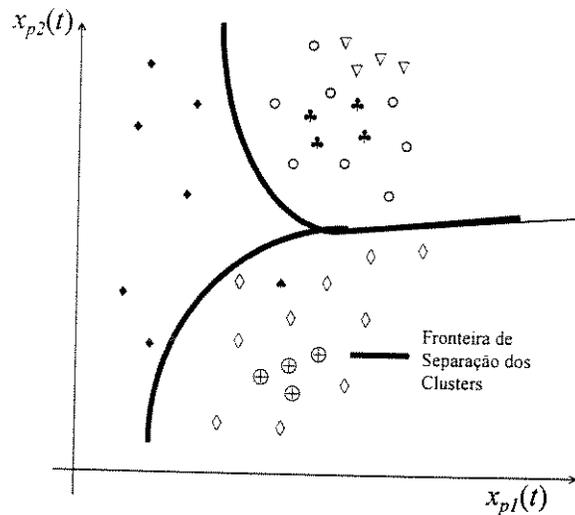


Fig. 1.7 - Ilustração de falhas separadas em cluster

Se o número de clusters e a sua respectiva composição não são conhecidos a priori, a rede neural auto-organizada com treinamento não supervisionado, mostrada no capítulo 2, pode ser utilizada no primeiro estágio. Para os sistemas de classificação do segundo estágio foram utilizadas as redes neural e *neurofuzzy and/or*.

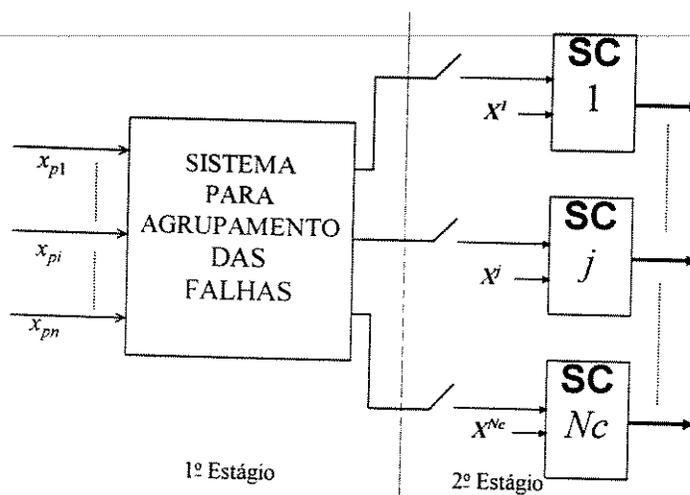


Fig. 1.8 - Sistema de Classificação distribuído

A estrutura mostrada na figura 1.1 foi usada para detectar e isolar falhas em sistemas dinâmicos para três estratégias diferentes. A diferença básica de uma para outra é a forma de geração do padrão, que é o vetor de entrada do sistema de classificação. Na primeira, o padrão é formado somente utilizando as entradas e saídas da planta. Já na segunda e terceira, chamadas de estratégias híbridas, são utilizados também resíduos obtidos via observadores de estados em modos deslizantes e identificação de parâmetros do modelo do sistema, respectivamente.

a) ESTRATÉGIA BASEADA EM ENTRADAS E SAÍDAS DA PLANTA E CLASSIFICAÇÃO DE PADRÕES

Para esta estratégia o padrão é gerado a partir dos vetores u , y e y^* (para sistema em malha fechada). Com estes vetores é feita a classificação da condição de operação da planta. A vantagem desta abordagem é que há a necessidade do modelo matemático analítico do sistema somente na fase de treinamento da rede, isto se resultados experimentais da planta não estão disponíveis (Caminhas *et alii*, 1996^{a,b}; Caminhas *et alii*, 1997).

b) ESTRATÉGIA BASEADA EM OBSERVADORES DE ESTADOS EM MODOS DESLIZANTES E CLASSIFICAÇÃO DE PADRÕES

Neste caso, são utilizados observadores de estados robustos baseados na teoria de modos deslizantes (sliding modes) (Takahashi e Peres, 1997), figura 1.9. Os observadores em modos deslizantes (OMD) são projetados para rejeitar perturbações paramétricas e entradas desconhecidas. Cada OMD rejeita perturbações em uma ou mais direções do espaço de estado. Empregando o OMD é possível obter um observador que seja robusto a um conjunto de falhas (erro de estimação nulo) e sensível as outras (erro diferente de zero). Comparando valores observados pelos “m” observadores é possível detectar as falhas no sistema.

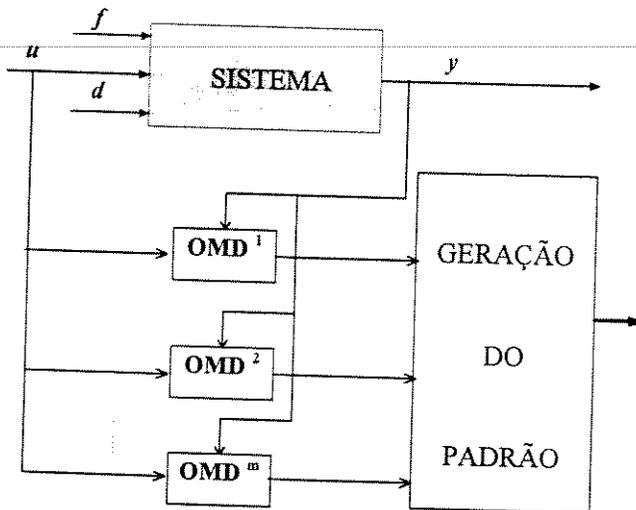


Fig. 1.9 - Geração de Padrão utilizando banco de observadores em modos deslizantes

c) ESTRATÉGIA BASEADA EM ESTIMAÇÃO DE PARÂMETROS E CLASSIFICAÇÃO DE PADRÕES

É empregada a topologia de rede *neurofuzzy*, baseada no neurônio *fuzzy* proposto por Yamakawa *et alii* (1992), para identificação dos parâmetros do modelo da planta, figura 1.10. Esses parâmetros são pesos sinápticos da rede. Os pesos W_0 (vetor) são determinados através do treinamento *off-line* da rede. Nesta etapa, os dados fornecidos para o treinamento são referentes a operações do sistema sem ocorrência de falhas. Já os pesos W são obtidos via

treinamento *on-line*. Quando ocorrerem falhas no sistema os pesos W serão diferentes de W_0 , sendo que essa diferença é usada para gerar os padrões do sistema de classificação.

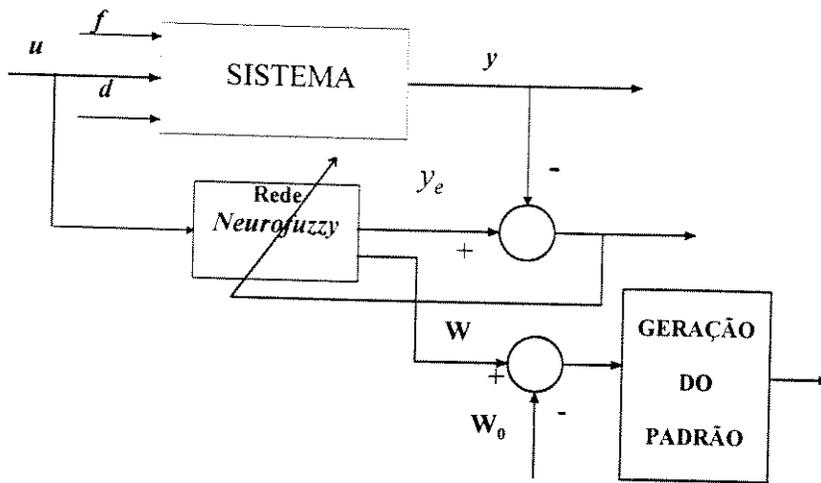


Fig. 1.10 - Geração de padrão utilizando identificação de parâmetros

1.2.2 - SISTEMAS DINÂMICOS PARA TESTE DAS ESTRATÉGIAS

Foram usados três sistemas dinâmicos para testar as estratégias descritas anteriormente, a saber: sistema de acionamento elétrico utilizando máquina de corrente contínua (C.C.); sistema de acionamento elétrico utilizando máquina de corrente alternada (C.A.) e sistema de tanques interativos.

a) SISTEMA DE ACIONAMENTO ELÉTRICO C.C.

O sistema de acionamento C.C. é composto por duas fontes de alimentação, conversores estáticos controlados, uma máquina de corrente contínua e uma carga mecânica, figura 1.11. Foram abordadas falhas nos conversores, na máquina e nos sensores de corrente e velocidade.

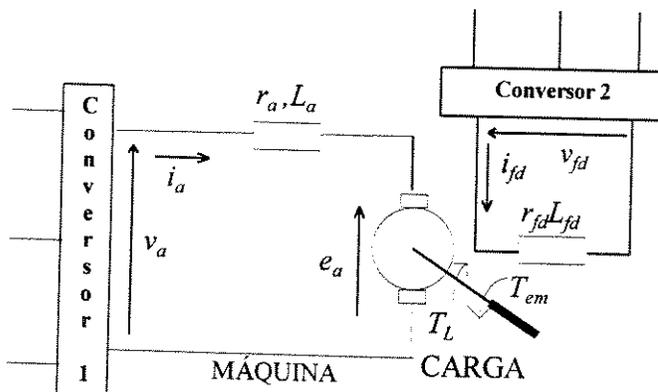


Fig. 1.11 - Representação do sistema de acionamento C.C.

b) SISTEMA DE ACIONAMENTO ELÉTRICO C.A.

O sistema de acionamento C.A., figura 1.12, é composto por uma fonte de alimentação (conversor estático ou rede elétrica), um motor de indução trifásico tipo rotor em gaiola e uma carga mecânica. Para este sistema foram analisadas as falhas de curto-circuito fase-neutro e abertura de fase, totalizando quatorze falhas, (Caminhas et alii, 1996^{a,b}).

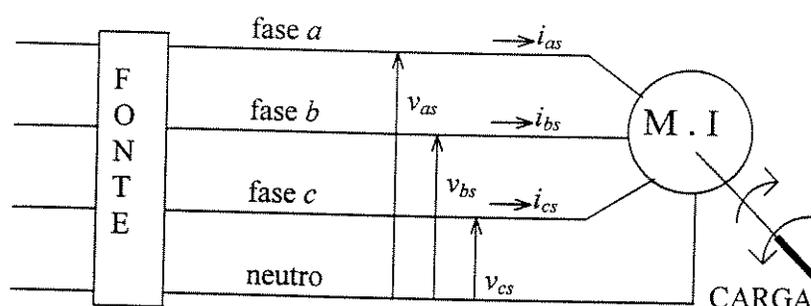


Fig. 1.12 - Sistema de acionamento C.A.

c) SISTEMA DE TANQUES INTERATIVOS

O terceiro sistema utilizado para validar as estratégias propostas é o Sistema de Tanques Interativos (ou acoplados)-STI, do Laboratório de Controle de Processos Industriais do CPDEE-UFMG (Jota et alii, 1995; Braga, 1994), figura 1.13. O objetivo é controlar o nível do tanque TQ-3, h_3 , e a sua vazão de saída, q_o , atuando nas válvulas FCV-1 e FCV-2. O controle do nível é feito a partir da atuação da válvula pneumática FCV-1, que controla a vazão de entrada de TQ-2, conseqüentemente controla a vazão de entrada de TQ-3. Já a vazão de saída de TQ-3 é controlada atuando na válvula pneumática FCV-2.

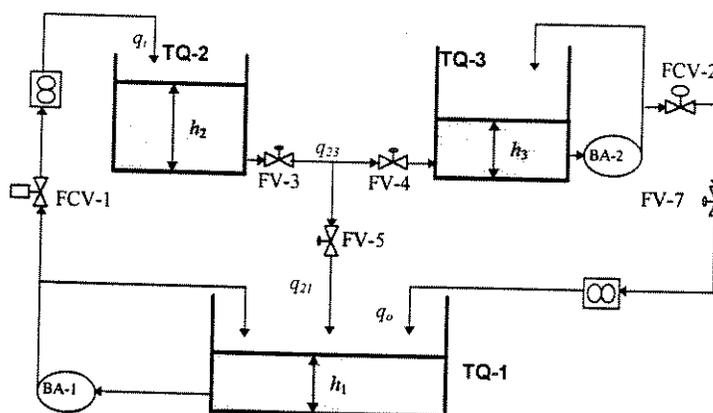


Fig. 1.13 - Fluxo de massa nos Tanques Interativos

Para esse sistema, foi testado o desempenho para falhas de vazamento, entupimento e falhas nas bombas, válvulas pneumáticas e sensores da planta. Foram utilizados o modelo e

alguns resultados experimentais obtidos por Pereira (1996), para validação do desenvolvimento (Caminhas *et alii*, 1997).

1.3 - ESTRUTURAÇÃO DO TEXTO

A tese é dividida em sete capítulos. No capítulo em questão as estratégias de detecção e diagnóstico de falhas em sistemas dinâmicos são descritas.

No capítulo 2 é estudada a aplicação de redes *neurofuzzy* e redes neurais em sistema de classificação de padrões. É proposta uma nova estrutura de rede *neurofuzzy*, bem como o seu algoritmo de treinamento. Esta rede é constituída de neurônios *fuzzy* tipo *and* e *or*, que são modelados por normas *s* e *t* (Pedrycz e Rocha, 1993). Com esta rede é possível implementar um conjunto de regras *fuzzy* do tipo se-então. Com relação à rede neural utilizada, esta é do tipo Kohonen com treinamento supervisionado (Kohonen, 1990). No algoritmo clássico de treinamento foi acrescentado mais um passo, que permite eliminar neurônios em excesso da rede. Além disso são testadas as normas N_1 , N_2 e N_∞ na determinação do neurônio vencedor. São apresentados resultados para quatro exemplos didáticos, permitindo uma análise comparativa das duas estruturas de rede.

No capítulo 3 são modelados os sistemas dinâmicos que serão utilizados para testar as estratégias de detecção e diagnóstico de falhas, descritos anteriormente: sistema de acionamento elétrico utilizando máquina de corrente contínua (C.C.); sistema de acionamento elétrico utilizando máquina de corrente alternada (C.A.) e sistema de tanques interativos. Os modelos matemáticos destes sistemas são não lineares e de ordem 3, 7 e 2, respectivamente. Tal desenvolvimento é necessário porque em todas as estratégias estudadas neste trabalho há necessidade de treinamento, o que normalmente exige um grande número de padrões para aprendizado correto.

No capítulo 4 são projetados observadores de estados para entradas e/ou parâmetros desconhecidos ou incertos, empregando a teoria de modos deslizantes. Este observador, como mostrado em Takahashi e Peres (1997), é uma generalização dos diversos tipos de observadores com entrada desconhecida presentes na literatura. A escolha pelo observador em modo deslizante recai no fato de simplicidade de projeto e excelente desempenho, já comprovado em trabalhos anteriores (Caminhas *et alii*, 1996^{d,e}).

No capítulo 5 é empregada uma rede *neurofuzzy* para identificação de sistemas dinâmicos não lineares. Essa rede *neurofuzzy* é constituída por neurônios *fuzzy* propostos por Yamakawa *et alii* (1992). A vantagem desta rede é que o seu treinamento é muitas vezes mais rápido que o treinamento da rede multicamadas tipo perceptron, como será mostrado. O fato para tal rapidez é que na rede *neurofuzzy* o erro é função quadrática dos pesos, além disso convexa, logo só existe um ponto de mínimo, sendo este global. Já para rede multicamada existem mínimos locais, o que dificulta a convergência para um erro exigido. Além do tempo de treinamento ser pequeno, o tempo de processamento também é pequeno. Isto é um fator importante quando da implementação em tempo real. Para a rede *neurofuzzy* são necessárias apenas operações básicas de soma e multiplicação.

No capítulo 6 estas ferramentas foram usadas em detecção e diagnóstico de falhas em sistemas dinâmicos. As estratégias propostas são baseadas: somente em sistema de

classificação de padrões ou em geração residual e sistema de classificação de padrões (estratégias híbridas). São apresentadas duas estruturas híbridas. Na primeira os resíduos são gerados a partir de comparações entre os estados para diversos observadores em modos deslizantes. Na segunda são utilizados parâmetros do modelo da planta, que são os pesos da rede *neurofuzzy*, capítulo 5. A utilização de sistema de classificação de padrões juntamente com geração residual é justificada pelo fato da lógica de determinação da falha a partir do resíduo não ser uma tarefa trivial. Com utilização da rede *neurofuzzy* proposta no capítulo 2, por exemplo, a lógica é determinada automaticamente, via treinamento. Os resultados de simulações apresentados permitem fazer avaliações da três estruturas.

Finalmente, no capítulo 7 são tiradas as conclusões do trabalho, ressaltando as principais contribuições, e sugeridas algumas proposta de continuidade.

CAPÍTULO 2

CLASSIFICAÇÃO DE PADRÕES USANDO LÓGICA FUZZY E REDE NEURAL

2.1 - INTRODUÇÃO

Considere um conjunto de m padrões $X_p = (x_{p1}, \dots, x_{pi}, \dots, x_{pn})$, $p = 1, 2, \dots, m$, onde x_{p1} a x_{pn} são as n coordenadas de X_p . O problema de classificação consiste agrupar os m padrões em N_c classes (1, 2, ..., N_c). Este tipo de problema tem aplicação em diversas áreas do conhecimento, tais como estatística, economia, engenharia, física, química, psicologia, biologia e outras. Para resolvê-lo podem ser utilizadas diversas abordagens, entre elas, redes neurais artificiais (Kohonen, 1990; Simpson, 1990; Kosko, 1992; Fausett, 1994; Anderson, 1995) e lógica *fuzzy* (Pedrycz, 1993; Mitra, 1994; Nozaki *et alii*, 1994^{a,b}; Genther *et alii*, 1994; Ishibuchi *et alii*, 1994; Chung e Lee, 1994; Mon, 1994; Leyeghi *et alii*, 1994; Abe e Lan, 1995; Figueiredo *et alii*, 1995). Aqui são apresentadas duas estruturas de sistema de classificação de padrões: a primeira, baseada em sistema *fuzzy*, implementada por uma rede *neurofuzzy* (Caminhas *et alii*, 1995^{a,b}; Caminhas *et alii*, 1996^c), e a segunda, baseada em uma rede neural.

A rede *neurofuzzy* proposta implementa um conjunto de regras “*fuzzy*” tipo *se-então*. Nesta topologia são empregados neurônios lógicos tipo “*and*” e “*or*” (Pedrycz e Rocha, 1993), que são modelados por normas e co-normas triangulares. O algoritmo de aprendizado da rede é responsável pela geração e modificação da base de regras, via treinamento.

Para efeito de comparação, é apresentado um outro sistema de classificação baseado em rede neural auto-organizada com algoritmo de treinamento tipo LVQ (“Learning Vector Quantization”) (Kohonen, 1990). Para esta rede, além da norma N_2 , são testadas as normas N_1 (norma 1) e N_∞ (norma infinita), como medidas de distância, para determinar o neurônio vencedor. O objetivo é a redução do tempo de processamento da rede. Além disso, é adotada uma política que permite eliminar os neurônios em excesso após o treinamento.

Estas duas estruturas foram testadas em exemplos didáticos de classificação de padrões, onde são ressaltados os passos dos algoritmos. Os resultados apresentados permitem uma análise comparativa das duas estruturas. Finalmente, no final do capítulo é feita uma análise da complexidade dos algoritmos de processamento das duas estruturas de sistema de classificação.

2.2 - CLASSIFICAÇÃO DE PADRÕES USANDO LÓGICA FUZZY

2.2.1 - SISTEMA FUZZY

Considere o sistema baseado em lógica *fuzzy*, ou simplesmente sistema *fuzzy*, representado na figura 2.1 (Wang, 1994). Esse sistema é constituído por uma base de regras e um mecanismo de inferência *fuzzy*. O sistema *fuzzy* faz o mapeamento $U \rightarrow V$ de um espaço de entrada para um espaço de saída, onde $U = U_1 \times \dots \times U_n \subset \mathbb{R}^n$ e $V \subset \mathbb{R}^1$, “ \times ” é o produto cartesiano.

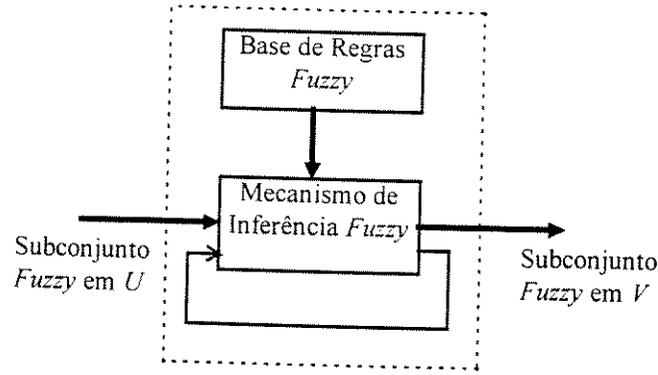


Fig. 2.1 - Configuração básica do sistema *fuzzy*

A base de regras é constituída de M regras dos tipo:

$$R^{(l)} : \text{se } x_1 \text{ e' } F_1^l \text{ e } \dots \text{ e } x_n \text{ e' } F_n^l, \text{ então } y \text{ e' } G^l \quad (2.1)$$

onde F_i^l e G^l são subconjuntos *fuzzy* definidos em $U_i \subset \mathfrak{R}$ e $V \subset \mathfrak{R}$, respectivamente, com $i = 1, 2, \dots, n$, $X = (x_1, \dots, x_n)^T \in U$ e $y \in V$ são as variáveis lingüísticas de entrada e saída, respectivamente, e $l = 1, 2, \dots, M$.

A regra $R^{(l)}$ é uma afirmação (ou proposição) condicional *fuzzy* denotada por $A \rightarrow B$, onde $A = F_1^l \times \dots \times F_n^l$ e $B = G^l$, sendo interpretada como uma relação *fuzzy* \mathbf{R}_l , que pode ser definida de várias formas, tais como:

$$\mu_{\mathbf{R}_l} = \mu_{A \rightarrow B}(X, y) = \mu_A(X) \wedge \mu_B(y), \text{ conjunção } \textit{fuzzy}; \quad (2.2)$$

$$\mu_{\mathbf{R}_l} = \mu_{A \rightarrow B}(X, y) = \mu_A(X) \vee \mu_B(y), \text{ disjunção } \textit{fuzzy}; \quad (2.3)$$

e outras (Lee, 1990). Para as expressões acima \wedge é um operador de interseção e \vee de união. Exemplos de operadores \wedge são as normas \mathbf{t}^1 (mínimo, produto e outros). Exemplos de \vee são as normas \mathbf{s}^2 (máximo, soma probabilística, soma limitada e outros).

A função $\mu_A(X)$ é calculada da seguinte forma:

$$\mu_A(X) = \mu_{F_1^l \times \dots \times F_n^l}(X) = \mathbf{T}_{i=1}^n \mu_{F_i^l}(x_i) \quad (2.4)$$

onde $\mathbf{T}_{i=1}^n y_i = (((y_1 \mathbf{t} y_2) \mathbf{t} y_3) \dots) \mathbf{t} y_n$

¹Uma norma \mathbf{t} é uma função $\mathbf{t}: [0, 1] \times [0, 1] \rightarrow [0, 1]$ tal que, $x, y, w \in [0, 1]$:

1. $x \mathbf{t} w \leq y \mathbf{t} z$, se $x \leq y, w \leq z$
2. $x \mathbf{t} y = y \mathbf{t} x$
3. $(x \mathbf{t} y) \mathbf{t} z = x \mathbf{t} (y \mathbf{t} z)$
4. $x \mathbf{t} 0 = 0; x \mathbf{t} 1 = x$

² Uma norma \mathbf{s} é uma função $\mathbf{s}: [0, 1] \times [0, 1] \rightarrow [0, 1]$, satisfazendo as propriedades (1) a (3) acima e ainda

5. $x \mathbf{s} 0 = x; x \mathbf{s} 1 = 1$

O mapeamento será realizado por uma única relação *fuzzy* que é uma combinação de todas as relações *fuzzy* provenientes das diversas regras. Esta combinação envolve um operador de agregação de regras:

$$R = \text{agreg}(R_1, R_2, \dots, R_M) \quad (2.5)$$

Usualmente, o operador de agregação é interpretado como um operador de união (utilizando a operação *max*), embora exista uma ampla classe de operadores de agregação (Gomide *et alii*, 1995).

Considere o seguinte problema: uma vez estabelecida a base de regras *fuzzy*, determine a saída “*y*” para uma entrada *X* conhecida. Este problema pode ser formulado da seguinte maneira:

- Fato: $X \text{ é } A'$
- Regra: Se ($X \text{ é } A$) então ($y \text{ é } B$)
- Consequência: $y \text{ é } B'$

que é o *modus ponens generalizado*. Considerando uma regra “*l*”, o subconjunto *fuzzy* B'_l , para essa regra, pode ser determinado a partir da regra da composição:

$$B'_l = A' \circ R_l \quad (2.6)$$

Em geral, a regra da composição é expressa na forma (Gomide *et alii*, 1995; Wang, 1994):

$$\mu_{B'_l}(y) = \sup_{X \in U} \{ \mu_{A'}(X) \mathbf{t} \mu_{R_l}(X, y) \} \quad (2.7)$$

um caso particular bastante usado é a composição *max-min*, proposta por Zadeh (1973). Neste caso *sup* e *t* correspondem aos operadores *max* e *min*, respectivamente.

Finalmente, a contribuição das *M* regras é calculada usando um operador de agregação:

$$B' = \text{agreg}(B'_1, B'_2, \dots, B'_l, \dots, B'_M) \quad (2.8)$$

Se for utilizada uma norma *s* como operador de agregação tem-se:

$$\mu_{B'}(y) = \mathbf{S}_{l=1}^M \mu_{B'_l} \quad (2.9)$$

onde $\mathbf{S}_{l=1}^m y_l = (((y_1 \mathbf{s} y_2) \mathbf{s} y_3) \dots) \mathbf{s} y_m$

A figura 2.2 ilustra o processo de inferência *max-min* quando existem 2 regras: $A_i \rightarrow B_i$ e $A_j \rightarrow B_j$. A' é o fato de entrada, representado como um subconjunto *fuzzy* (Gomide *et alii*, 1995).

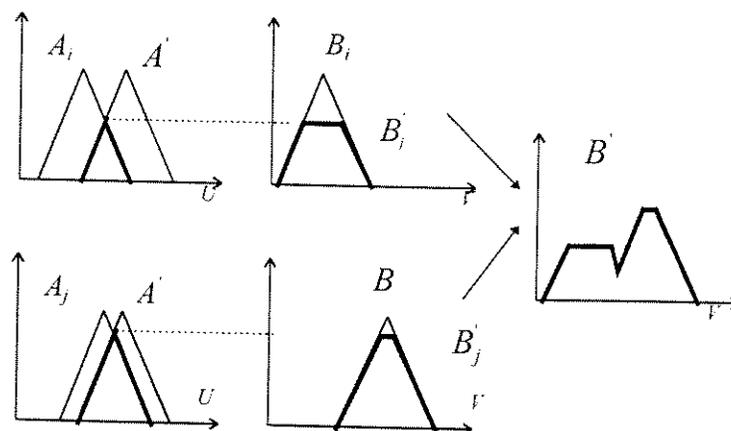


Fig. 2.2 - Ilustração do mecanismo de inferência *fuzzy* (*max-min*)

Neste trabalho, serão usadas as normas **t** e **s** para obter:

$$\mu_{R^{(i)}} = \mu_{A \rightarrow B}(X, y) = \mu_A(X) \mathbf{t} \mu_B(y) \quad (2.10)$$

$$\mu_{B_i'}(y) = \max_{X \in U} \{ \mu_{A'}(X) \mathbf{t} \mu_{R_i}(X, y) \} \quad (2.11)$$

$$\mu_{B'}(y) = \mathbf{S}_{i=1}^M \mu_{B_i'} \quad (2.12)$$

2.2.2 - CLASSIFICAÇÃO DE PADRÕES USANDO SISTEMA FUZZY

2.2.2.1 - Formulação do problema

Dado um padrão $X_p = (x_{p1}, \dots, x_{pi}, \dots, x_{pn})$, o problema de classificação consiste em determinar a classe (1, 2, ..., Nc) a qual o padrão pertence. Esse problema pode ser tratado como sendo um mapeamento $U \rightarrow V$, de um espaço de entrada U para um espaço de saída V . Portanto, o desenvolvimento do item anterior pode ser usado para resolvê-lo. Para os casos tratados neste trabalho, as entradas (coordenadas do padrão) e saída (índice da classe do padrão) são variáveis que assumem valor exato, são *fuzzy singleton*, ou seja, $A' = X_p$ e $B = 1, 2, \dots, Nc$. Como as entradas e saída do sistema *fuzzy* mostrado na figura 2.1 são subconjuntos *fuzzy*, as etapas de *fuzzificação* e *defuzzificação* são necessárias, figura 2.3.

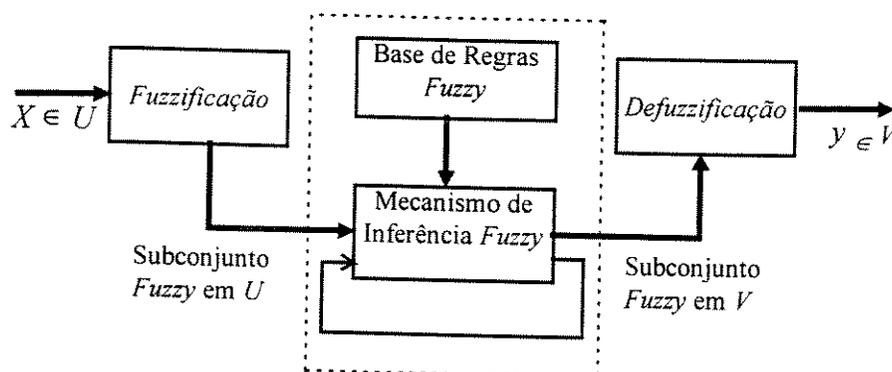


Fig. 2.3 - Configuração básica do sistema *fuzzy* para classificação de padrões

Na etapa de *fuzzificação* os valores exatos das entradas são transformados em subconjuntos *fuzzy*. Na *defuzzificação* é feita a operação inversa para obter a saída, que corresponde à classe do padrão, que é um valor inteiro positivo. Neste caso, o processo de inferência fica bastante simplificado, pois, considerando as propriedades das normas **s** e **t** e a definição de *fuzzy singleton* (Wang, 1994) as expressões de (2.10) a (2.12) ficam:

$$\mu_{R_i}(X, y) = \mu_{A \rightarrow B}(X, y) = \mu_A(X_p) = \prod_{i=1}^n \mu_{F_i'}(x_{pi}) \quad (2.13)$$

$$\mu_{B_i}(y) = \mu_{R_i}(X, y) \quad (2.14)$$

$$\mu_{B'}(y) = \mathbf{S}_{i=1}^M \left[\prod_{i=1}^n \mu_{F_i'}(x_{pi}) \right] \quad (2.15)$$

A expressão (2.13) corresponde à etapa de *fuzzificação*. Para a *defuzzificação*, como o valor de “ y ” é inteiro positivo, será utilizado o critério do máximo (*max*), ou seja:

$$y = \arg \left\{ \max_{y \in V} [\mu_{B'}(y)] \right\} \quad (2.16)$$

O critério de *defuzzificação* baseado no centro de área não pode ser utilizado neste caso, porque poderia resultar num valor de “ y ” que não seja inteiro. A ilustração do mecanismo de inferência para o sistema *fuzzy* mostrado na figura 2.3 será feita a partir do exemplo descrito a seguir.

2.2.2.2 - Exemplo de Problema de Classificação de Padrões

A figura 2.4 mostra um exemplo de um problema de classificação apresentado em (Nozaki *et alii*, 1994^{a,b}). O problema consiste em classificar 20 padrões $X_p = (x_{p1}, x_{p2})$, $p=1,2,\dots,20$, como classe “1” para os padrões representados por (O) e “2” para (Δ).

• Caracterização do problema

Para este exemplo, os conjuntos U e V são definidos por:

$$U = \left\{ (x_{p1}, x_{p2}) \mid 0 \leq x_{p1} \leq 1 \text{ e } 0 \leq x_{p2} \leq 1 \right\} \subset \mathbb{R}^2 \quad (2.17)$$

$$V = \{1, 2\} \quad (2.18)$$

Serão considerados dois subconjuntos *fuzzy* F_1 (pequeno) e F_2 (grande) que são caracterizados pelas funções de pertinência mostradas na figura 2.4.

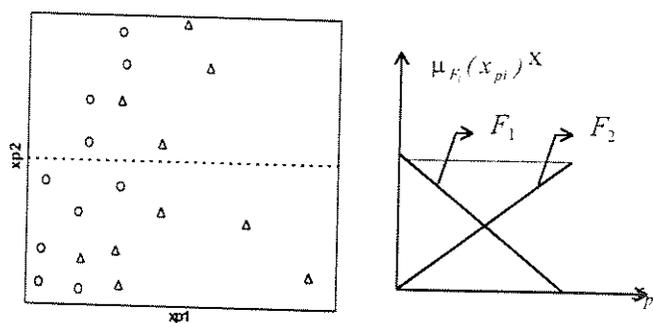


Fig. 2.4 - Exemplo de problema de classificação de padrões.

- **Montagem da base de regras**

Para facilitar o raciocínio, considere que as coordenadas x_{p1} e x_{p2} são divididas, de forma igual, em duas partições *fuzzy*, definidas pelos subconjuntos *fuzzy* pequeno, F_1 , e grande F_2 , figura 2.4. Observando a figura 2.4 nota-se que para x_{p1} pequeno e x_{p2} pequeno a quantidade de padrões tipo 'O', classe 1, é maior que os do tipo "Δ", classe 2. Continuando este raciocínio pode-se montar as seguintes regras *fuzzy* para o problema:

- (CRI): *se* (x_{p1} é F_1) *e* (x_{p2} é F_1) *então* y é 1;
se (x_{p1} é F_1) *e* (x_{p2} é F_2) *então* y é 1;
se (x_{p1} é F_2) *e* (x_{p2} é F_1) *então* y é 2;
se (x_{p1} é F_2) *e* (x_{p2} é F_2) *então* y é 2.

- **Mecanismo de inferência fuzzy**

Utilizando as expressões (2.13) a (2.16) o mecanismo de inferência para (CRI) pode ser representado de forma esquemática conforme mostrado na figura 2.5. A agregação dos antecedentes é modelada pela norma **t** e a agregação das regras pela norma **s**.

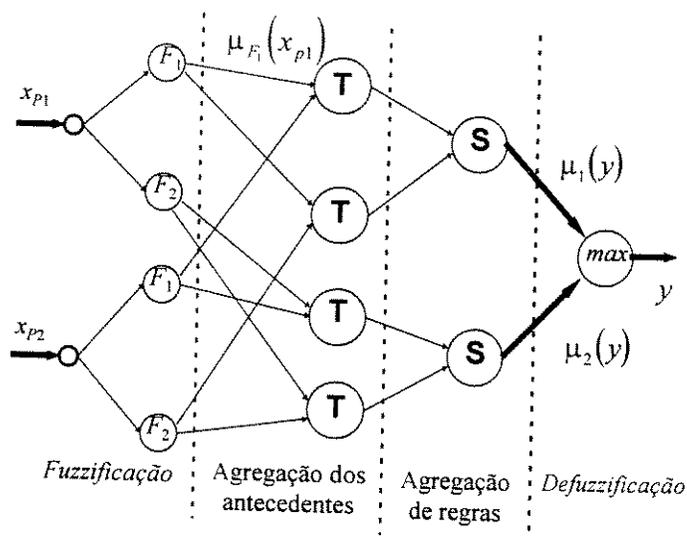


Fig. 2.5 - Representação esquemática do mecanismo de inferência *fuzzy* para o (CRI)

Para exemplificar o esquema mostrado na figura 2.5, considere os operadores *min* e *max* como normas **t** e **s**, respectivamente, e o padrão $X_p=(0.4; 0.3)$. A figura 2.6 ilustra as etapas do mecanismo de inferência *fuzzy* para este padrão. O resultado da classificação é que o padrão pertence à classe 1 com grau de pertinência 0.6 e à classe 2 com grau de pertinência 0.4. Utilizando o *max* como critério de *defuzzificação*, pode-se concluir que a saída “*y*” é igual a “1”.

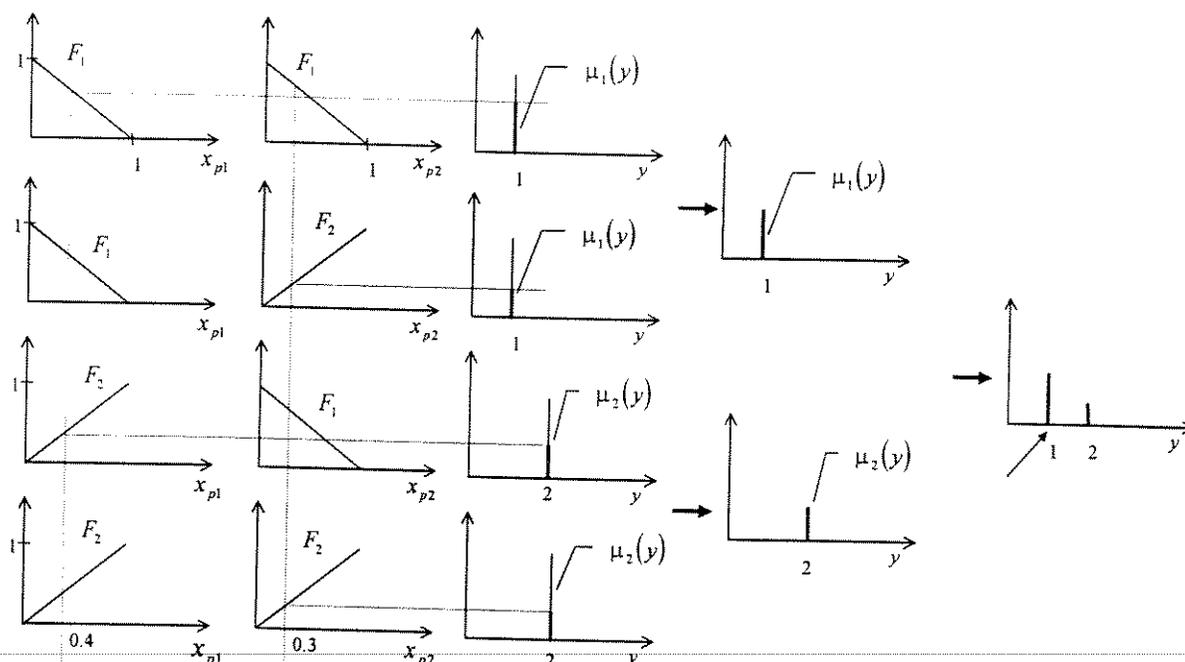


Fig. 2.6 - Mecanismo de inferência *fuzzy* para o padrão padrão $X_p=(0,4; 0,3)$

Com o procedimento esquematizado na figura 2.6 é possível chegar à superfície que separa as duas classes, mostrada na figura 2.7. Observa-se que com (CR1) não é possível classificar corretamente todos os padrões.

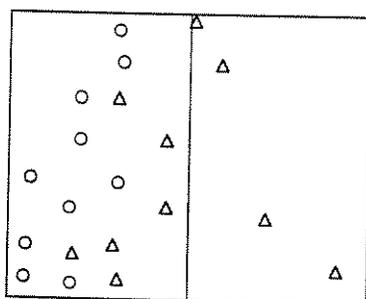


Fig. 2.7 - Superfície de separação usando o conjunto de regras (CR1)

Analisando o mecanismo de inferência *fuzzy*, mostrado na figura 2.6, nota-se que as funções de pertinência, que caracterizam os subconjuntos *fuzzy* das variáveis de entrada, influenciam o resultado da classificação. Técnicas baseadas em ajuste de parâmetros dessas funções de pertinência, utilizando métodos de aprendizado supervisionado a partir de exemplos, são apresentadas, por exemplo, em Wang (1994) e Pedrycz (1993). Para tal, estas técnicas utilizam algoritmo de treinamento tipo *backpropagation*. Uma outra técnica, proposta

por Nozaki *et alii* (1994^{a,b}), trabalha com funções de pertinência fixas e introduz grau de certeza nas regras, da seguinte forma:

- (CR2): *se* (x_{p1} é F_1) *e* (x_{p2} é F_1) *então* y é 1 com grau de certeza v_1 ;
se (x_{p1} é F_1) *e* (x_{p2} é F_2) *então* y é 1 com grau de certeza v_2 ;
se (x_{p1} é F_2) *e* (x_{p2} é F_1) *então* y é 2 com grau de certeza v_3 ;
se (x_{p1} é F_2) *e* (x_{p2} é F_2) *então* y é 2 com grau de certeza v_4 .

A vantagem dessa estrutura com relação às outras duas é que a forma de ajuste dos parâmetros (graus de certeza) é mais simples, já que o algoritmo não utiliza derivadas do erro com relação aos parâmetros. Nesse algoritmo o ajuste é feito a partir de treinamento competitivo, ou seja, se o padrão é classificado corretamente o grau de certeza da regra responsável pela classificação é aumentado. A figura 2.8 mostra a superfície de separação, usando essa abordagem, após ajustar os graus de certeza, v_i .

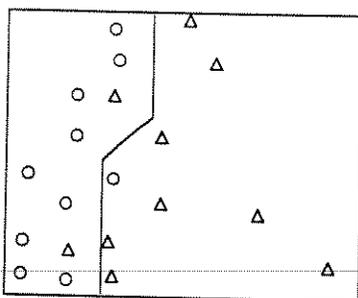


Fig. 2.8 - Superfície de separação usando o conjunto de regras (CR2) obtida a partir do programa implementado.

2.3 - IMPLEMENTAÇÃO DE SISTEMA DE CLASSIFICAÇÃO FUZZY VIA REDE NEUROFUZZY

Nos últimos anos o potencial de manuseio de incerteza proporcionado pela lógica *fuzzy* vem sendo combinado com *redes neurais artificiais*, que possuem características de aprendizagem e adaptação (Figueiredo, 1997). Esta simbiose vem gerando novas classes de *redes neurofuzzy* (Figueiredo, 1997; Keller *et alii*, 1992; Lin e Lee, 1991; Lin e Gunningham, 1995). Em (Figueiredo, 1997) são descritas algumas destas estruturas de *redes neurofuzzy*, que se diferenciam pela arquitetura, métodos de aprendizagem, parâmetros predefinidos e representação do conhecimento. Em Caminhas *et alii* (1995^a) é proposta uma estrutura de *rede neurofuzzy* para implementação de sistema de classificação *fuzzy*. Essa rede é constituída de neurônios lógicos de agregação tipo *and* e *or* (Pedrycz e Rocha, 1993), por isso será chamada de *rede neurofuzzy and/or*. Ela se enquadra naquelas estruturas onde é possível a inserção e extração de conhecimento na forma de regras *fuzzy* do tipo *se-então*, estabelecendo um mapeamento entre regras *fuzzy* e o conhecimento presente em sua arquitetura. A estrutura da *rede neurofuzzy and/or*, bem como seu algoritmo de treinamento, herdaram as seguintes características: capacidade de generalização dos neurônios *and* e *or*, no que diz respeito a utilização de diversas normas **s** e **t**; capacidade de extrair o conhecimento a partir de dados de entrada e saída, como ocorre nas estruturas propostas em (Nozaki *et alii*, 1994^{a,b}; Lin e Gunningham, 1995) e simplicidade do algoritmo de aprendizagem, sem utilizar derivadas, a exemplo da proposta por Nozaki *et alii* (1994^{a,b}).

2.3.1 - NEURÔNIOS LÓGICOS

Em Pedrycz e Rocha (1993) são propostos os neurônios lógicos de agregação *and* e *or*, mostrados na figura 2.9. Esses neurônios fazem o mapeamento $[0,1]^n \rightarrow [0,1]$. O parâmetro $w_i \in [0,1]^n$ representa a intensidade da conexão (peso) da entrada x_i no mapeamento.

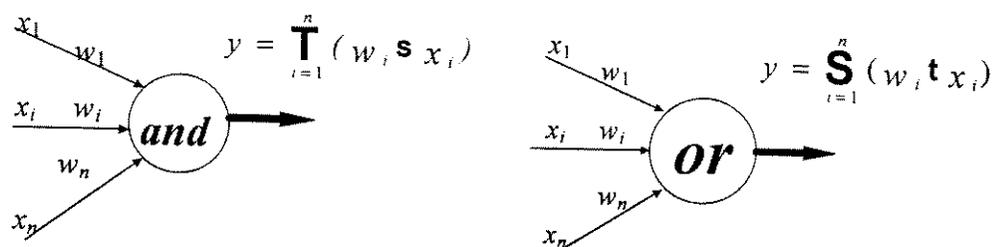


Fig. 2.9 - Neurônios lógicos de agregação

Considerando as propriedades referente às condições de contorno para as normas **t** e **s**, fazendo $w_i=0$ para o neurônio *and* e $w_i=1$ para o neurônio *or*, estes neurônios resultam nos operadores **t** e **s** utilizados na figura 2.5. Portanto, a estrutura mostrada na figura 2.5 pode ser generalizada pela rede lógica *neurofuzzy and/or* da figura 2.10. Para essa rede se $w_i=0$ para o neurônio *and* e usando os operadores *produto* e *max* para as normas **t** e **s**, respectivamente, tem-se a estrutura proposta por Nozaki *et alii* (1994^{a,b}). Neste caso, os pesos das conexões dos neurônios tipo *or* são equivalentes aos graus de certeza de (CR2).

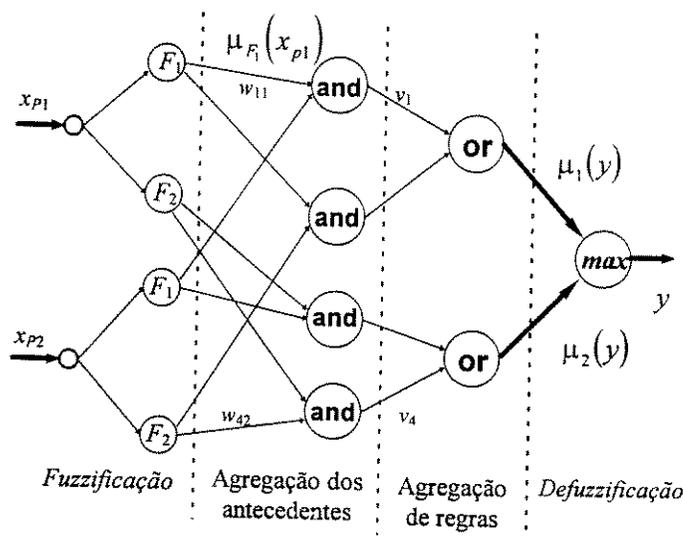


Fig. 2.10 - Topologia da rede *neurofuzzy and/or* para o exemplo 1.

A figura 2.11 mostra a superfície de separação obtida com a rede da figura 2.10. Nota-se que o número de classificações incorretas é menor do que os resultados obtidos utilizando (CR1) e (CR2).

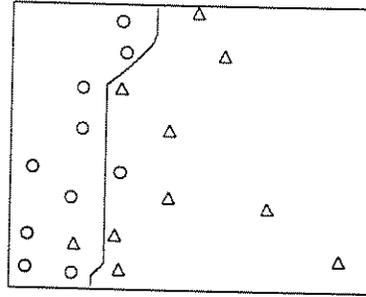


Fig. 2.11 - Superfície de separação usando rede *neurofuzzy and/or* obtida a partir do programa implementado.

2.3.2 - REDE NEUROFUZZY AND/OR

A rede *neurofuzzy and/or* pode ser generalizada para permitir a classificação de padrões com “ n ” coordenadas de entrada, N_c classes de saída e número de subconjuntos *fuzzy* diferente para cada variável de entrada, figura 2.12.

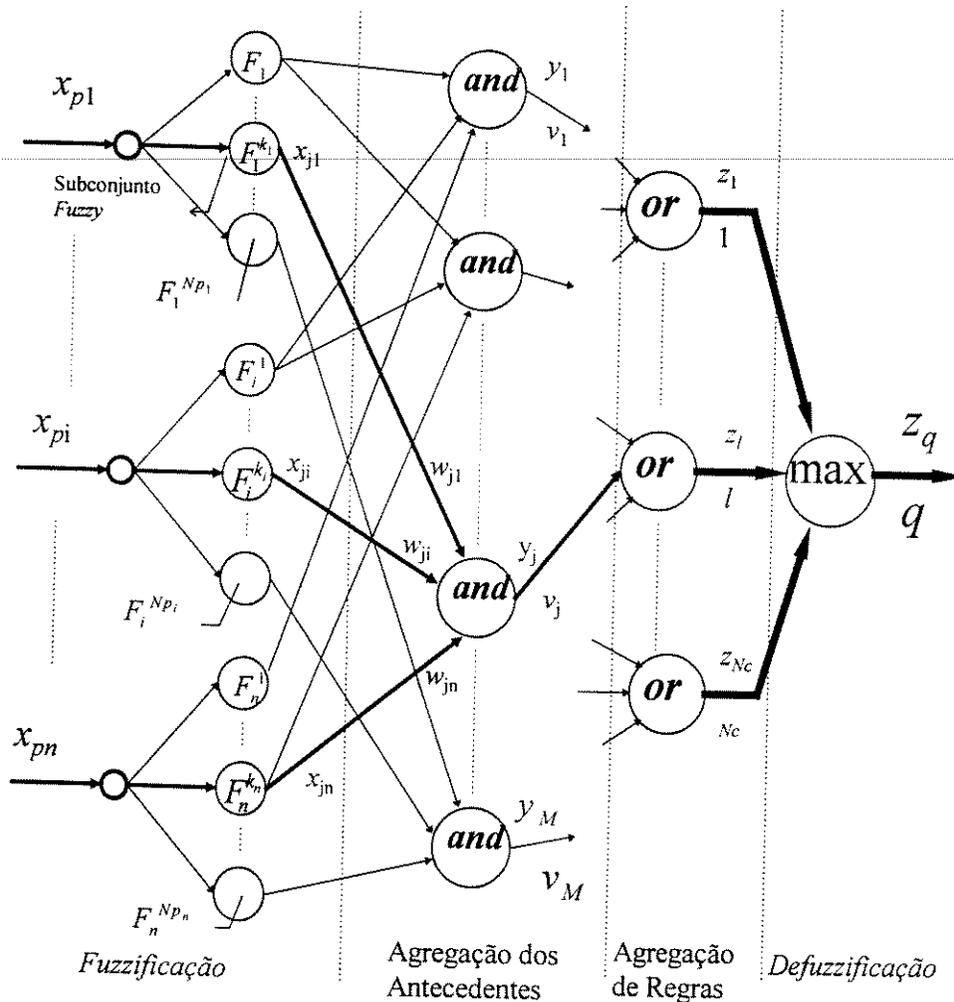


Fig. 2.12 - Rede *neurofuzzy and/or*

Com a rede *neurofuzzy and/or* da figura 2.12 é possível implementar um conjunto de regras $\mathbf{R} = \{R^{(j)}, j = 1, 2, \dots, M\}$ do tipo:

R^j : **Se** (x_{p1} é $F_1^{k_1}$ com peso da conexão= w_{j1}) ... **e** (x_{pi} é $F_i^{k_i}$ com peso da conexão= w_{ji}) ...
e (x_{pn} é $F_n^{k_n}$ com peso da conexão= w_{jn})

então o X_p pertence a classe l com peso da conexão= v_j .

Com relação à rede *neurofuzzy and/or*, tem-se:

- Np_i é o número de subconjuntos *fuzzy* da entrada “ i ”;
- “ j ” é o índice que indexa um neurônio tipo *and*. Da forma como foi estruturada a rede, o índice “ j ” é determinado a partir dos índices dos subconjuntos *fuzzy* das coordenadas de entrada, k_i , da seguinte maneira:

$$j = f(K) = k_n + \sum_{i=2}^n (k_{n-i+1} - 1) \cdot \left(\prod_{j=1}^i Np_{i+1-j} \right) \quad (2.19)$$

sendo K o vetor que contém os índices dos subconjuntos *fuzzy*, ou seja, $K = (k_1 \ \dots \ k_i \ \dots \ k_n)$.

Para exemplificar, suponha que: x_{p1} é F_1^1 ; x_{p2} é F_2^3 e x_{p3} é F_3^2 e $Np_1 = Np_2 = Np_3 = 3$. Neste caso, tem-se: $k_1 = 1$; $k_2 = 3$ e $k_3 = 2$

$$\therefore j = 2 + (k_2 - 1) \cdot N_3 + (k_1 - 1) \cdot N_3 N_2 = 2 + (3 - 1) \cdot 3 + (1 - 1) \cdot 9 = 8$$

- x_{p1} a x_{pn} são as n coordenadas do padrão X_p , entradas da rede;
- $x_{ji} = \mu_{F_i^{k_i}}(x_{pi})$ é grau de pertinência da variável x_{pi} na regra j ao subconjunto *fuzzy* $F_i^{k_i}$;
- w_{ji} é o peso da conexão da entrada “ i ” para o neurônio *and* indexado por j ;
- y_j é o valor da saída do neurônio *and* indexado por j , obtido usando a expressão:

$$y_j = \mathbf{T} \left(\sum_{i=1}^n w_{ji} \mathbf{s} x_{ji} \right) \quad (2.20)$$

- v_j é o peso da conexão da entrada “ j ” para o neurônio *or* indexado por q ;
- z_l é o valor da saída do neurônio *or* indexado por l , que corresponde ao grau de pertinência do padrão X_p à classe l , obtido usando a expressão:

$$z_l = \mathbf{S} \left(\sum_{j \in \mathbf{N}_l} v_j \mathbf{t} y_j \right) \quad (2.21)$$

onde \mathbf{N}_l é o conjunto dos índices dos neurônios tipo *and* que estão conectados no neurônio tipo *or* indexado por “ l ”;

- z_q é o valor que corresponde ao maior grau de pertinência e q é a classificação *crisp* do padrão, obtida a partir da *defuzzificação*;

2.4 - APRENDIZADO DA REDE NEUROFUZZY AND/OR

Assim como as redes neurais clássicas, a rede *neurofuzzy and/or* representada na figura 2.12 possui a capacidade de aprendizado. Será adotado um algoritmo de treinamento semelhante ao LVQ (Linear Vector Quantization) (Kohonen, 1990), para ajustar os pesos da rede. Vale ressaltar que esse treinamento é supervisionado, ou seja, dado um padrão X_p qualquer, sabe-se a priori qual é sua classificação. Por outro lado, é um treinamento competitivo, não utilizando derivadas para ajustar os pesos das conexões, conforme algoritmo 1.

• Algoritmo 1: Estrutura geral do algoritmo de treinamento da rede *neurofuzzy and/or*

1. gere as funções de pertinência;
2. gere as conexões da rede e inicialize os pesos w_{ji} e v_j ;
3. Processo Iterativo:
 - 3.1. apresente um padrão p à rede (normalmente escolhido de forma aleatória);
 - 3.2. determine os neurônios *and* e *or* ativos;
 - 3.3. efetue a *fuzzificação*;
 - 3.4. determine a classe vencedora;
 - 3.5. atualize os pesos w_{ji} e v_j ;
 - 3.6. teste a condição de parada (todos os padrões classificados corretamente ou número de iterações alcançado):
 - se é satisfeita vá para o passo 4;
 - se não é satisfeita volte ao passo 3.1.
4. fim.

O detalhamento deste algoritmo é apresentado a seguir.

2.4.1 - GERAÇÃO DAS FUNÇÕES DE PERTINÊNCIA

Serão discutidas duas formas de geração das funções de pertinência quais sejam: geração de funções triangulares normais, uniformemente distribuídas, e funções normais não uniformemente distribuídas. Para os dois tipos, as funções de pertinência são complementares, ou seja, a soma de duas funções sucessivas é igual a "1".

2.4.1.1 - Geração das Funções de Pertinência Tipo 1

Neste caso, têm-se funções de pertinência como as representadas na figura 2.13. As funções de pertinência são definidas pelos parâmetros x_{imin} e x_{imax} . O centro da função de pertinência indexada por " r " é a_{ir} , que corresponde ao ponto onde a função possui o valor máximo.

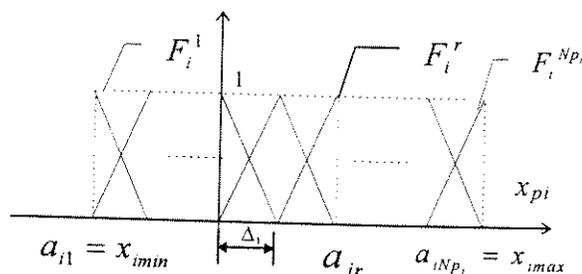


Fig. 2.13 - Funções de Pertinência tipo 1

Da figura 2.13, tem-se que:

$$\Delta_i = \frac{X_{imax} - X_{imin}}{Np_i - 1} \quad (2.22)$$

Com isto, o cálculo de uma função de pertinência r é feito da seguinte forma:

$$\mu_{F_i^r}(x_{pi}) = \begin{cases} \frac{(x_{pi} - a_{ir})}{\Delta_i} + 1, & a_{ir-1} \leq x_{pi} \leq a_{ir} \\ -\frac{(x_{pi} - a_{ir})}{\Delta_i} + 1, & a_{ir} < x_{pi} \leq a_{ir} \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.23)$$

2.4.1.2 - Geração das Funções de Pertinência Tipo 2

A utilização de funções de pertinência do tipo 1 pode não ser interessante em situações onde há concentração de padrões em determinadas regiões e dispersão em outras. Isso poderia exigir uma utilização de um número grande de subconjuntos *fuzzy*. Uma opção é trabalhar com funções não uniformemente distribuídas. Com isto, nas regiões de baixa concentração poder-se-ia trabalhar com um número de subconjuntos *fuzzy* menor. Uma maneira de fazer a geração automática das funções de pertinência é utilizar um algoritmo de *clusterização*, onde os centros dos *clusters* correspondam aos centros das funções de pertinência, a_{ir} , figura 2.14. O algoritmo de *clusterização* utilizado é baseado em rede neural auto-organizada (Kohonen, 1990), conforme figura 2.15. São empregadas n estruturas, uma para cada coordenada do padrão. Os pesos das redes correspondem aos valores dos centros das funções de pertinência. O número de neurônios na segunda camada (Np_i^0), que é um dado definido a priori, corresponde ao número de subconjuntos *fuzzy* para a coordenada “ i ”. Ao final do treinamento da rede este número pode ser reduzido.

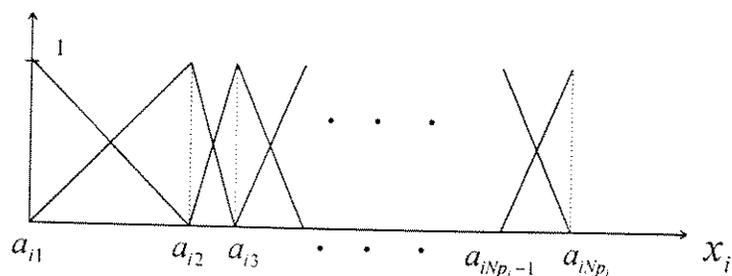


Fig. 2.14 - Funções de Pertinência tipo 2

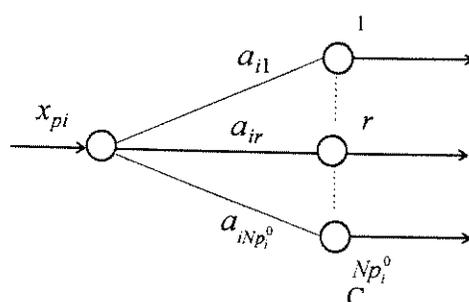


Fig. 2.15 - Rede neural auto-organizada para determinar os centros das funções de pertinência

O treinamento da rede neural auto-organizada da figura 2.15 é não supervisionado e competitivo (Kohonen, 1990). Somente o peso da conexão do neurônio vencedor é ajustado. Ao final do treinamento, aqueles neurônios que tiveram um baixo índice de desempenho, isto é, venceram poucas vezes, são eliminados da rede. Este procedimento permite determinar o número de funções de pertinência adequado, Np_i , e seus respectivos centros.

Algoritmo 2: Treinamento da rede neural auto-organizada para geração das funções de pertinência

1. inicializações:

- pesos a_{ir} :

$$a_{i1} = x_{imin} \quad (2.24)$$

$$a_{ir} = a_{i(r-1)} + \Delta_i, \text{ para } r = 2, 3, \dots, Np_i^0 \quad (2.25)$$

- índice de desempenho: $id_i(r) = 0$, para $r = 1, 2, \dots, Np_i^0$, ou seja, as funções de pertinência iniciais são do tipo 1. Este procedimento de inicialização geralmente proporciona uma convergência mais rápida que a inicialização aleatória;

2. processo iterativo:

- 2.1. apresente um padrão p à rede e atualize o peso da conexão do neurônio vencedor da seguinte forma:

$$a_{iL}(p+1) = a_{iL}(p) + \alpha(p) \cdot [x_{pi} - a_{iL}(p)] \quad (2.26)$$

onde L é o índice do neurônio vencedor, que é aquele cujo peso da conexão possui o valor mais próximo de x_{pi} , ou seja:

$$L = \arg \left\{ \min_r |x_{pi} - a_{ir}| \right\} \quad (2.27)$$

2.2. reduzir o passo $\alpha(p)$;

2.3. atualize o índice de desempenho do neurônio vencedor, fazendo:

$$Id_i(L) = Id_i(L) + 1 \quad (2.28)$$

2.4. verifique o teste de parada ($a_{ij}(k) - a_{ij}(k) \leq \varepsilon$):

- se não é satisfeito volte ao passo 2.1;
- se é satisfeito vá para o passo 3.

3. elimine todos os neurônios cujo valor de Id_i seja menor que um limiar N (inteiro positivo). Seja Nne_i o número de neurônios eliminados para a coordenada “ i ”. Portanto, o número de subconjuntos *fuzzy* para a coordenada i é:

$$Np_i = Np_i^0 - Nne_i \quad (2.29)$$

4. fim.

Uma vez definidos o número de funções de pertinência e seus respectivos centros, o cálculo das mesmas pode ser feito com um número reduzido de operações, utilizando o conjunto de parâmetros (a_{ir} , α_{eir} e α_{dir}), onde:

$$\alpha_{eir} = \frac{1}{a_{ir} - a_{ir-1}} \quad (2.30)$$

$$\alpha_{dir} = \frac{1}{a_{ir} - a_{ir+1}} \quad (2.31)$$

são as derivadas da função de pertinência à esquerda e direita de a_{ir} , respectivamente. Com estes parâmetros o cálculo das funções de pertinência é bastante simples, envolvendo duas operações de soma e uma de multiplicação.

$$\mu_{F_i^r}(x_{pi}) = \begin{cases} \alpha_{eir} \cdot (x_{pi} - a_{ir-1}) + 1, & a_{ir-1} \leq x_{pi} \leq a_{ir} \\ \alpha_{dir} \cdot (a_{ir+1} - x_{pi}) + 1, & a_{ir} < x_{pi} \leq a_{ir+1} \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.32)$$

2.4.2 - GERAÇÃO DAS CONEXÕES DA REDE E INICIALIZAÇÃO DOS PESOS w_{ji} e v_j

Neste passo são estabelecidas as conexões entre os neurônios *and* e os neurônios *or*. Este passo equivale à geração das regras *se-então*. Existem algumas propostas de geração automática encontradas na literatura (Wang e Mendel, 1992; Li e Wu, 1994; Nozaki *et alii*, 1994^{a,b}; Abe e Lan, 1995). A estratégia que será adotada é uma adaptação da idéia proposta por Wang e Mendel (1992) e Nozaki *et alii* (1994^{a,b}). As etapas do algoritmo são:

Algoritmo 3: Geração das conexões da rede *neurofuzzy and/or*

Para $j=1$ até M faça:

a) geração das conexões

compute β_l para $l=1,2,\dots,Nc$

$$\beta_l = \sum_{x_p \in \text{Classe } l} \prod_{i=1}^n x_{ji} \quad (2.33)$$

determine a classe q tal que

$$q = \arg \left\{ \max_l \{ \beta_1, \dots, \beta_l, \dots, \beta_{Nc} \} \right\} \quad (2.34)$$

Isto define a conexão do neurônio *and* indexado por “ j ” ao neurônio *or* indexado por “ q ”. Se $\beta_q = 0$ ou se existir mais de um máximo para β_q , então o neurônio *and* “ j ” não conecta a nenhuma classe de saída.

b) Inicialize os pesos w_{ji} e v_j , fazendo:

$$w_{ji} = 0, \forall i = 1, 2, \dots, n \text{ e } j = 1, 2, \dots, M \quad (2.35)$$

$$v_j = \frac{\beta_q - \sum_{l \neq q} \beta_l / (Nc - 1)}{\sum \beta_l} \quad (2.36)$$

Quanto maior a diferença de β_q para as demais classes, mais próximo é o valor de v_j de “1”. Quando existirem duas, ou mais, classes dominantes o valor de v_j é próximo de “0”. A inicialização dos pesos v_j utilizando a expressão acima implica em redução do tempo de processamento do algoritmo, pois melhora a superfície de separação, como mostrado na figura 2.16. A linha pontilhada corresponde à superfície com os pesos v_j inicializados iguais a 1 e a linha contínua corresponde ao cálculo dos pesos v_j a partir da expressão (2.36).

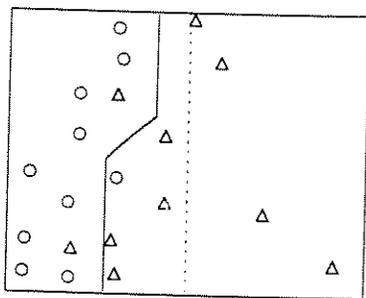


Fig. 2.16 - Superfície de separação usando a rede *neurofuzzy*: inicializando $v_j=1$ (linha pontilhada) e inicializando v_j usando a expressão (2.36) (linha contínua).

2.4.3 - DETERMINAÇÃO DOS NEURÔNIOS *AND* E *OR* ATIVOS

Considerando funções de pertinência como as mostradas nas figura 2.13 e 2.14, para cada padrão apresentado os respectivos graus de pertinência são diferentes de zero no máximo para dois subconjuntos *fuzzy*. Estes serão definidos como subconjuntos ativos, que por sua vez definem os neurônios *and* ativos. Com isto, dos M neurônios *and*, no máximo 2^n serão ativos. Estes neurônios são determinados da seguinte maneira:

- dado um padrão $X_p = (x_{p1}, \dots, x_{pi}, \dots, x_{pn})$,

sejam $K^1 = (k_1^1 \ \dots \ k_i^1 \ \dots \ k_n^1)$ o vetor que contém os índices da primeira função de pertinência, para cada coordenada do padrão, diferente de zero e $K^2 = (k_1^2 \ \dots \ k_i^2 \ \dots \ k_n^2)$ um vetor tal que:

$$K_i^2 = \begin{cases} k_i^1 + 1, & \text{se } \mu_{k_i^1}(x_{pi}) \neq 1 \\ k_i^1, & \text{caso contrário} \end{cases} \quad (2.37)$$

O número de neurônios tipo *and* ativo, N_a , é igual a 2^{Pa} , que é no máximo igual a 2^n , sendo Pa número de elementos tal que $k_i^1 \neq k_i^2$, para $i=1,2,\dots,n$. Estes neurônios podem ser determinados a partir da combinação dos vetores K^1 e K^2 , como ilustrado no exemplo:

- suponha uma rede com 3 entradas definidas no intervalo $[-2, +2]$ e considere 3 subconjuntos *fuzzy* para cada entrada, figura 2.17. Para um determinado padrão p de coordenadas $X_p=(0.5, -2, 1)$, tem-se: para $x_{p1}=0.5$ os subconjuntos *fuzzy* ativos são F_1^2 e F_1^3 ; para $x_{p2}=-2$ somente o subconjuntos *fuzzy* F_2^1 é ativo, e para $x_{p3}=1$ os subconjuntos *fuzzy* ativos são F_3^2 e F_3^3 , conforme mostrada na figura 2.17. Definindo assim os vetores $K^1 = (2, 1, 2)$ e $K^2 = (3, 1, 3)$.

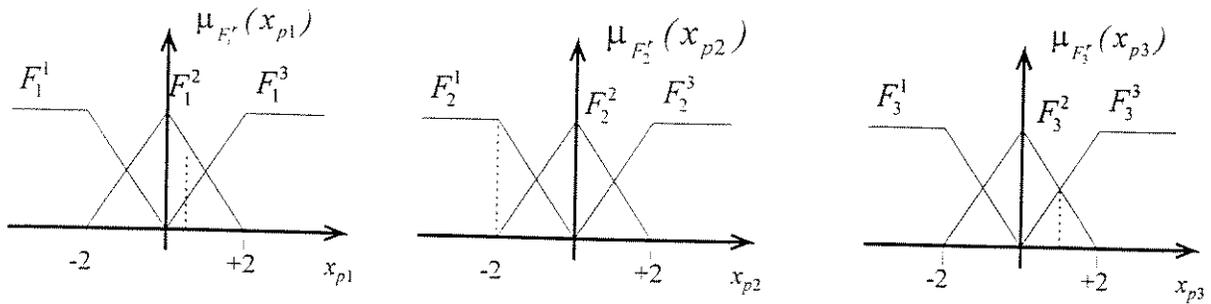


Fig. 2.17 - Subconjuntos ativos

Combinando os vetores $K^1 = (2, 1, 2)$ e $K^2 = (3, 1, 3)$ chega-se aos seguintes vetores K^3 : $(2, 1, 2)$; $(2, 1, 3)$; $(3, 1, 2)$ e $(3, 1, 3)$, que, a partir da expressão (2.19), permitem identificar os neurônios *and* indexados 'j' igual a 11, 12, 20 e 21 como neurônios ativos.

Observe que, do conjunto de 27 neurônios *and*, somente $N_a = 2^{Pa} = 2^2 = 4$ são ativos. A vantagem do emprego desse passo do algoritmo é que somente os neurônios ativos serão considerados para efeito de cálculo nos passos seguintes. Com isto, o tempo de processamento do algoritmo fica independente do número de partições do espaço de entrada, ou seja, independe do número de neurônios *and*.

Seja L^* o conjunto dos neurônios *and* ativos. Este conjunto e as conexões estabelecidas no passo 2 determinam o conjunto Q^* que indexa os neurônios *or* ativos.

2.4.4 - FUZZIFICAÇÃO

Na etapa de fuzzificação são feitos os cálculos dos graus de pertinência correspondentes às coordenadas de entrada, somente para os subconjuntos ativos definidos em K^1 . No caso em que $K_i^1 \neq K_i^2$ calcula-se também $\mu_{k_i^2}(x_{pi})$. Como as funções de pertinência são complementares, tem-se:

$$\mu_{k_i^2}(x_{pi}) = 1 - \mu_{k_i^1}(x_{pi}) \quad (2.38)$$

Vale ressaltar que é necessário calcular no máximo "2n" graus de pertinência.

2.4.5 - DETERMINAÇÃO DA CLASSE VENCEDORA

Para determinar a classe vencedora faça:

(i) Compute $Z_l \forall l \in Q^*$ (somente para os neurônios ativos)

$$z_l = \mathbf{S}_{j \in l} (v_j \mathbf{t} y_j) \quad (2.39)$$

$$y_j = \mathbf{T}_{i=1}^n (w_{ji} \mathbf{S} x_{ji}) \quad (2.40)$$

O valor de Z_l , uma vez normalizado (dividido pela soma de todos os Z_l , para $l=1,2,\dots, Nc$), fornece o grau de pertinência do padrão X_p à classe l .

(ii) *Defuzzificação* para determinar a classe vencedora

O índice da classe vencedora, q , é determinado a partir da *defuzzificação*. Utilizando operador *max* como critério tem-se:

$$z_q = \max_{l \in \mathcal{Q}^*} \{z_l\} \quad (2.41)$$

2.4.6 - ATUALIZAÇÃO DOS PESOS

O processo de atualização dos pesos das conexões da rede é do tipo punição ou recompensa. Quando a rede classificar corretamente o padrão os pesos das conexões mais significativas são aumentados. Caso contrário, eles são reduzidos. Portanto, a atualização dos pesos w_{ij} e v_j é feita da seguinte maneira:

Seja q o índice da classe vencedora, determinada no passo 6.

Se q é também o índice da classe correta, então faça:

$$v_J(k+1) = v_J(k) + \alpha_1 \cdot [1 - v_J(k)] \quad (2.42)$$

$$w_{Jl}(k+1) = w_{Jl}(k) + \alpha_2 \cdot [1 - w_{Jl}(k)] \quad (2.43)$$

senão faça:

$$v_J(k+1) = v_J(k) - \alpha_3 \cdot v_J(k) \quad (2.44)$$

$$w_{Jl}(k+1) = w_{Jl}(k) - \alpha_4 \cdot w_{Jl}(k) \quad (2.45)$$

sendo “ I ” e “ J ” os indexadores das conexões mais significativas para a classificação, sendo determinados da seguinte maneira:

$$J = \arg \left\{ \max_j \left\{ [v_j \mathbf{T} y_j] \mid \forall j \in L^* \text{ e } X_p \in \text{Classe } q \right\} \right\} \quad (2.46)$$

$$I = \arg \left\{ \min_i \left\{ [x_{ji} \mathbf{S} w_{ji}] \mid \forall i = 1, 2, \dots, n \text{ e } j = J \right\} \right\} \quad (2.47)$$

Para as expressões de (2.42) a (2.45) $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ são parâmetros de treinamento tais que $0 < \alpha_1 < \alpha_3 < 1$ e $0 < \alpha_2 < \alpha_4 < 1$ (Nozaki *et alii*, 1994^a).

2.5 - CLASSIFICAÇÃO DE PADRÕES USANDO REDE NEURAL

2.5.1 - ESTRUTURA DA REDE NEURAL

Como foi visto no item 2.2.2, o problema de classificação de padrões nada mais é do que um mapeamento de um espaço de entrada para um espaço de saída. As redes neurais artificiais vêm sendo utilizadas com sucesso em problemas desse tipo. Entre as várias topologias de redes neurais, a estrutura que será usada neste trabalho é a rede auto-organizada, mostrada na figura 2.18, com algoritmo de treinamento tipo LVQ (Linear Vector Quantization) (Kohonen, 1990).

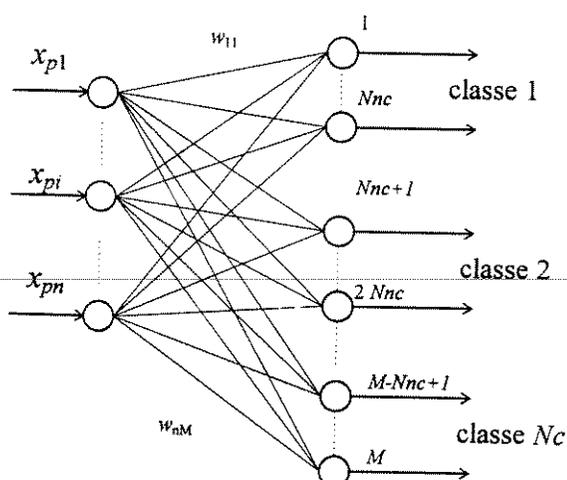


Fig. 2.18 - Rede neural auto-organizada para classificação de padrões

Essa estrutura é uma generalização da rede da figura 2.15. Neste caso, a rede possui n neurônios de entradas e $M = N_c \cdot N_{nc}$ neurônios de saída, sendo N_c o número de classes e N_{nc} o número de neurônios por classe. Como, geralmente, não se sabe o número de neurônios necessários para cada classe, o procedimento usado para eliminação dos neurônios em excesso é o mesmo adotado para obtenção das funções de pertinência, ou seja, eliminar aqueles neurônios com índice de desempenho abaixo de um determinado limiar. A matriz de pesos da rede $W = [W^1 \ W^2 \ \dots \ W^{M-1} \ W^M]$ onde $W^l = [w_{1l} \ w_{2l} \ \dots \ w_{n-l} \ w_{nl}]^T$ é o vetor coluna " l " de W . Dado um padrão X_p a sua classificação é feita da seguinte maneira:

$$q = \text{int}[(L-1) / L] + 1 \quad (2.48)$$

onde $\text{int}[*]$ representa a parte inteira de $[*]$ e L o índice da coluna W^L , tal que a **distância** entre W^L e X_p seja mínima. Como medida de **distância** as normas consideradas foram:

$$N_1: d_1^L = \sum_{i=1}^n |x_{pi} - w_{iL}| \quad (2.49)$$

$$N_2: d_2^L = \left(\sum_{i=1}^n (x_{pi} - w_{iL})^2 \right)^{1/2} \quad (2.50)$$

$$N_\infty: d_\infty^L = \max_{1 \leq i \leq n} |x_{pi} - w_{iL}| \quad (2.51)$$

Do ponto de vista de tempo de processamento, tp , para cálculo da norma tem-se a seguinte relação: $tp(N_2) > tp(N_1) > tp(N_\infty)$. Isto pode ser justificado pelo número de operações envolvidas.

2.5.2 - ALGORITMO DE APRENDIZADO DA REDE NEURAL AUTO-ORGANIZADA

O algoritmo de treinamento da rede auto-organizada para classificação de padrões é supervisionado do tipo LVQ (Kohonen, 1990), algoritmo 4.

Algoritmo 4: Treinamento da rede neural auto-organizada para classificação de padrões

1. inicialização: pesos (normalmente é feito de forma aleatória) e índice de desempenho (=0);
2. processo iterativo:
 - 2.1. apresente um padrão p à rede, determine o neurônio vencedor, L , determine a classe vencedora, q , expressão (2.48) e atualize os pesos do vetor coluna W^L da seguinte forma:
 - se a rede classifica corretamente o padrão

$$\text{então } W^L(k+1) = W^L(k) + \alpha(p) \cdot [X_p(k) - W^L(k)] \quad (2.52)$$
 - senão

$$W^L(k+1) = W^L(k) - \alpha(p) \cdot [X_p(k) - W^L(k)] \quad (2.53)$$
 - 2.2. reduzir o passo $\alpha(p)$;
 - 2.3. atualize o índice de desempenho do neurônio vencedor se a classificação foi correta, fazendo:

$$Id_i(L) = Id_i(L) + 1 \quad (2.54)$$
 - 2.4. verifique o teste de parada (se todos os padrões foram classificados corretamente ou se número máximo de iterações foi atingido):
 - se não é satisfeito volte ao passo 2.1;
 - se é satisfeito vai para o passo 3.
3. elimine os neurônios cujo valor de Id_i seja menor que N (inteiro positivo).
4. fim.

2.6 - EXEMPLOS DIDÁTICOS DE CLASSIFICAÇÃO DE PADRÕES

Os sistemas de classificação apresentados nos itens anteriores foram testados para os exemplos mostrados na figura 2.19. Para permitir uma melhor visualização dos resultados, em todos os exemplos os padrões possuem duas coordenadas, definidas no intervalo $[0,1]$ e duas classes. Os parâmetros de treinamento utilizados nos exemplos foram $\alpha_1=\alpha_2 = 0.1$ e $\alpha_3=\alpha_4 = 0.001$ e o número de iterações foi 500.

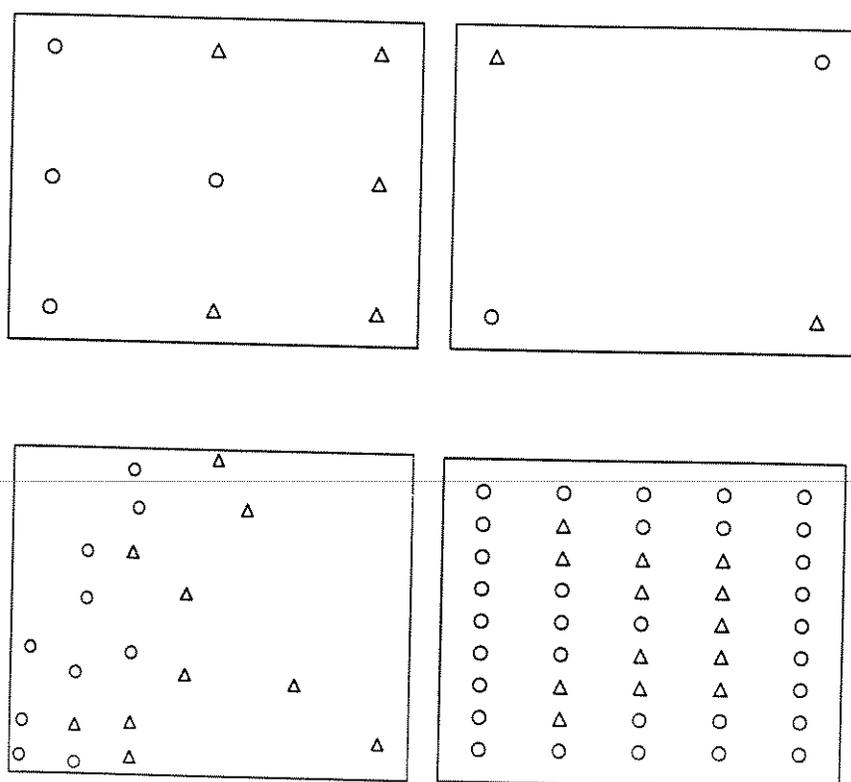


Fig. 2.19 - Exemplos para testar os sistemas de classificação

2.6.1 - RESULTADOS PARA A REDE NEUROFUZZY AND/OR

2.6.1.1 - Exemplo 1

A figura 2.20 mostra as superfícies de separação com as coordenadas de entrada x_{p1} e x_{p2} divididas em duas partições *fuzzy*. A linha traço fino se refere à rede com os pesos w_{ji} mantidos em zero. Nota-se que, com somente duas partições, não é possível classificar corretamente todos os padrões. Já para a mesma rede e considerando que os pesos w_{ji} são atualizados durante o treinamento é possível classificar todos os padrões corretamente, conforme linha de traço grosso da figura 2.20.

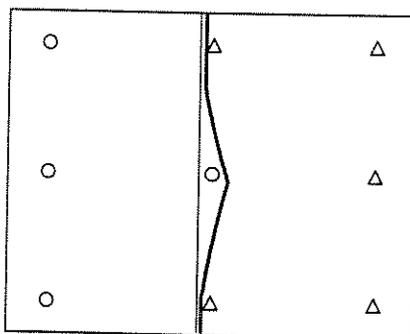


Fig. 2.20 - Superfície de separação para o exemplo 1: rede *neurofuzzy*; $N_{p1}=N_{p2}=2$; S-norma = max ; t-norma = produto; $w_{ij}=0$ (linha de traço fino) e $w_{ij} \neq 0$ (linha de traço grosso).

Detalhando os passos do algoritmo de treinamento da rede *neurofuzzy and/or*, tem-se:

Passo 1: Geração das funções de pertinência tipo 1:

Para gerar as funções de pertinência é preciso determinar os parâmetros Δ_i , x_{imin} e x_{imax} , para $i=1,2$. Os p padrões, no total de 9, para o exemplo apresentam as seguintes coordenadas, tabela 1:

Tabela 1: Padrões para o exemplo 1.

p	x_{p1}	x_{p2}	Classe
1	0.1	0.1	1
2	0.1	0.5	1
3	0.1	0.9	1
4	0.5	0.1	2
5	0.5	0.5	1
6	0.5	0.9	2
7	0.9	0.1	2
8	0.9	0.5	2
9	0.9	0.9	2

onde classe 1: (O) e classe 2: Δ .

Da tabela pode-se extrair os seguintes valores:

$$x_{1min} = 0.1; \quad x_{1max} = 0.9; \quad x_{2min} = 0.1; \quad \text{e} \quad x_{2max} = 0.9;$$

Com isto, tem-se

$$\Delta_1 = \frac{X_{1max} - X_{1min}}{N_s - 1} = \frac{0.9 - 0.1}{2 - 1} = 0.8 \quad \text{e} \quad \Delta_2 = \frac{X_{2max} - X_{2min}}{N_s - 1} = \frac{0.9 - 0.1}{2 - 1} = 0.8$$

Finalmente, as funções de pertinência para as coordenadas x_{p1} e x_{p2} são iguais, mostradas na figura 2.21.

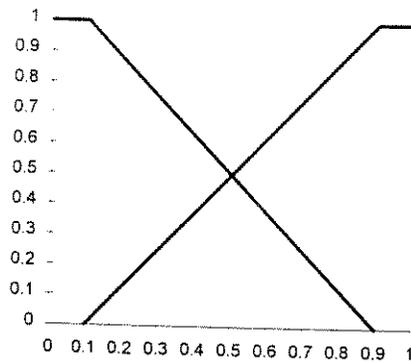


Fig. 2.21 - Funções de pertinência para as coordenadas x_{p1} e x_{p2} .

Passo 2: Geração das conexões da rede e inicialização dos pesos

Para $j=1$ até 4 (M) tem-se os seguintes valores para β_1 e β_2 , tabela 2:

Tabela 2: Valores de β_1 para o exemplo 1 e $N_{p1} = N_{p2}=2$

x_{p1}	x_{p2}	j expressão (2.19)	vetor K	β_1	β_2	"q" índice do máximo entre β_1 e β_2
F_1^1	F_2^1	1	[1 1]	1.75	0.5	1
F_1^1	F_2^2	2	[1 2]	1.75	0.5	1
F_1^2	F_2^1	3	[2 1]	0.25	2	2
F_1^2	F_2^2	4	[2 2]	0.25	2	2

Com estes valores de β_1 e β_2 são definidas as conexões entre os neurônios da rede, figura 2.10, mostrada anteriormente.

- **inicialização dos pesos:**

Com os valores de β_1 e β_2 e utilizando a expressão (2.36) determina-se os pesos v_j : $v_1=0.56$; $v_2=0.56$; $v_3=0.78$ e $v_4=0.78$. Os pesos w_{ij} são inicializados em zero.

- **Etapa de treinamento :**

Suponha que o padrão $p=2$ é apresentado à rede:

Como as coordenadas de X_2 são (0.1, 0.5), os vetores K^1 e K^2 assumem os valores: $K^1 = (1,1)$ e $K^2 = (1,2)$. Isto define que os neurônios ativos são 1 e 2. Para $X_3 = (0.1, 0.9)$, $K^1 = (1,2)$ e $K^2 = (1,2)$, ativando o neurônio 2. Já para $X_5 = (0.5, 0.5)$, $K^1 = (1,1)$ e $K^2 = (2,2)$, ativando os neurônios 1,2,3 e 4.

- **Análise da influência do número de partições *fuzzy***

Para este exemplo 1 mesmo fazendo $w_{ij} = 0$ com 3 partições *fuzzy* para as coordenadas x_{p1} e x_{p2} , representadas pelas funções de pertinência mostradas na figura 2.22, é possível classificar corretamente sem a necessidade de treinamento da rede, conforme mostrado na figura 2.23.

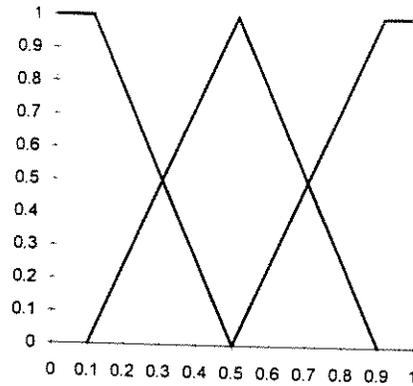


Fig. 2.22 - Funções de pertinência para as coordenadas x_{p1} e x_{p2} , respectivamente.

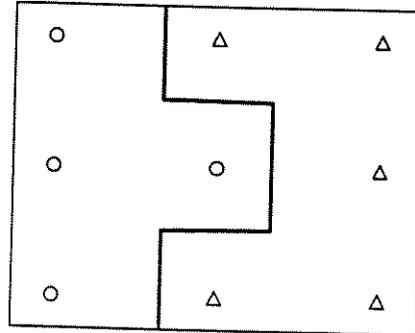


Fig. 2.23 - Superfície de separação para o exemplo 1: rede *neurofuzzy and/or* com $N_{p1}=N_{p2}=3$.

- **Influência do tipo de s-norma e t-norma**

Para os resultados apresentados até agora a s-norma adotada foi o máximo e a t-norma o produto. Para avaliar a influência da escolha destas normas são mostrados resultados para s-norma = *max* e t-norma = *min*, figura 2.24; s-norma = *soma probabilística* e t-norma = *min*, figura 2.25; s-norma = *soma probabilística* e t-norma = *produto*, figura 2.26. Pode-se notar que a *soma probabilística* e *produto* como s-norma e t-norma respectivamente, proporcionam superfícies de separação não-linear, ou seja, não são segmentos de reta. Já para as demais combinações a superfície é formada por segmentos de retas. O preço que se paga utilizando a soma probabilística é o número de operações maior.

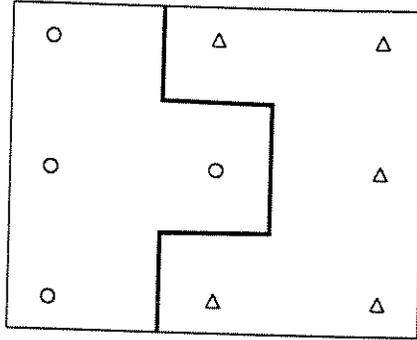


Fig. 2.24 - Superfície de separação para o exemplo 1:
rede neurofuzzy and/or; $N_{p1}=N_{p2}=3$;
S-norma = *max* ; t-norma = *min* e $w_{ij}=0$.

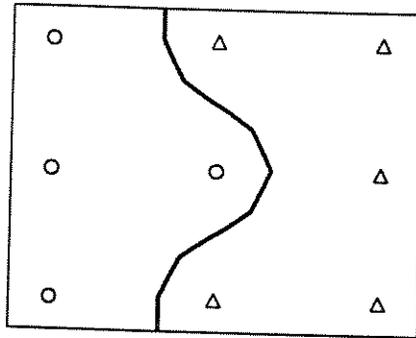


Fig. 2.25 - Superfície de separação para o exemplo 1:
rede neurofuzzy and/or; $N_{p1}=N_{p2}=3$;
S-norma = *soma probabilística* ; t-norma = *min* e $w_{ij}=0$.

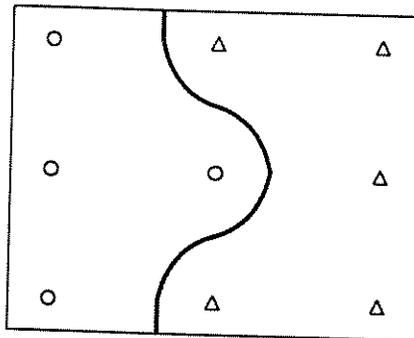


Fig. 2.26 - Superfície de separação para o exemplo 1:
rede neurofuzzy and/or; $N_{p1}=N_{p2}=3$;
S-norma = *soma probabilística* ; t-norma = *produto* e $w_{ij}=0$.

2.6.1.2 - Exemplo 2

A figura 2.27 mostra o resultado obtido para o exemplo 2. Neste caso foram necessárias apenas duas partições *fuzzy* para cada coordenada e somente com a inicialização dos pesos é possível classificar corretamente os padrões.

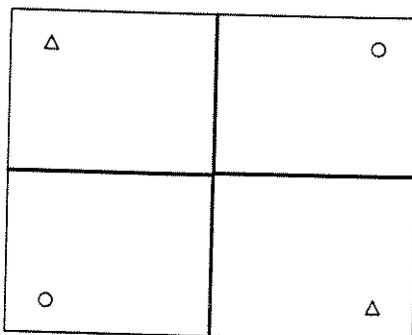


Fig. 2.27 - Superfície de separação para o exemplo 2:
neurofuzzy and/or; $N_{p1}=N_{p2}=2$;

2.6.1.3 - Exemplo 3

A figura 2.28 mostra o resultado, após o treinamento, para o exemplo 3. As funções de pertinência são do tipo 2. O número inicial de funções foi 6, sendo que, após o treinamento, três delas foram eliminadas para a coordenada x_{p2} . A s-norma foi a *soma probabilística* e t-norma o *produto*.

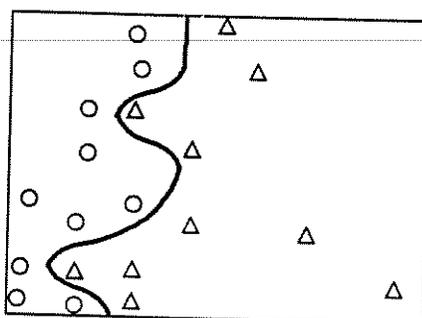


Fig. 2.28 - Classificação para o exemplo 3:
rede neurofuzzy and/or; $N_{p1}=6$ e $N_{p2}=3$;
s-norma = soma probabilística e t-norma = produto.

2.6.1.4 - Exemplo 4

As figuras 2.29 e 2.30 mostram os resultados para o exemplo 4 com as coordenadas de entradas divididas em 5 partições *fuzzy*, funções de pertinência tipo 1; para a figura 2.29 a s-norma = *soma probabilística* e t-norma = *produto*. Já para a figura 2.30 a s-norma = *max* e t-norma = *produto*. Mais uma vez fica evidenciada a não-linearidade da superfície de separação quando é utilizada como s-norma a *soma probabilística* e t-norma o *produto* e a linearidade (separação por segmentos de reta) para o *max* como norma **s** independente da norma **t**. Nas figuras a linha de traço fino corresponde à superfície antes do treinamento da rede e a linha de traço grosso após o treinamento.

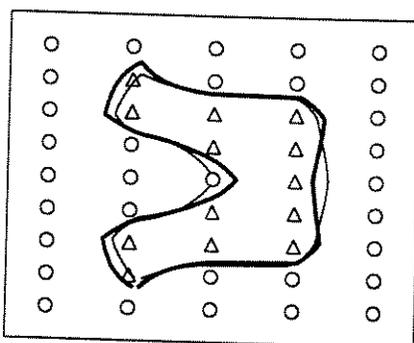


Fig. 2.29 - Classificação para o exemplo 4: rede *neurofuzzy and/or*; $N_{p1}=N_{p2}=5$; s-norma = soma probabilística e t-norma = produto.

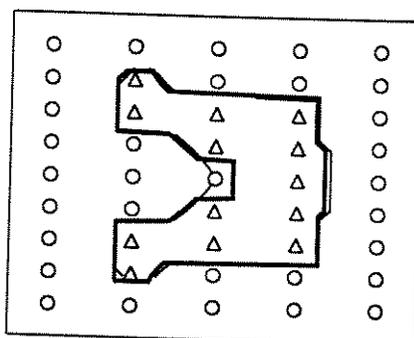


Fig. 2.30 - Classificação para o exemplo 4: rede *neurofuzzy and/or*; $N_{p1}=N_{p2}=5$; s-norma = max e t-norma = produto.

2.6.2 - RESULTADOS PARA A REDE NEURAL

2.6.2.1 - Exemplo 1

Os mesmos exemplos foram testados para rede neural. A figura 2.31 mostra a superfície de separação para a configuração final, ou seja, após o treinamento. Os pesos da rede foram inicializados aleatoriamente. A rede inicialmente tinha 25 neurônios para cada classe e após o treinamento ficou com 4 neurônios para a classe 1 (o) e 5 para a classe 2 (Δ). A norma utilizada para determinar o neurônio vencedor foi a N_2 . Após o treinamento os pesos dos neurônios que não foram eliminados assumiram os valores:

Tabela 3: Pesos da rede após o treinamento para o exemplo 1.

Índice do neurônio "j"	w_{1j}	w_{2j}	índice da classe
1	0.099	0.898	1
6	0.498	0.503	1
16	0.099	0.098	1
17	0.099	0.499	1
33	0.898	0.100	2
35	0.901	0.500	2
37	0.900	0.898	2
43	0.500	0.901	2
47	0.500	0.987	2

Como se pode observar, os valores dos pesos convergiram praticamente para os valores das coordenadas dos padrões, veja a tabela 1.

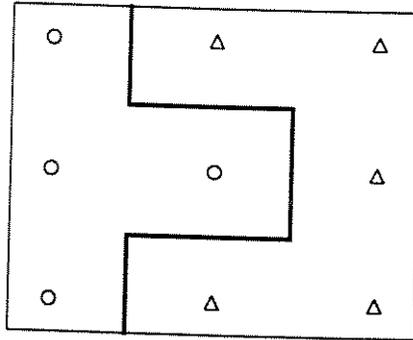


Fig. 2.31 - Superfície de separação para o exemplo 1: rede neural; norma N_2 ; rede inicia com 25 neurônios por classe e rede final com 4 neurônios para a classe 1 e 5 para a classe 2.

2.6.2.2 - Exemplo 2

A figura 2.32 mostra o resultado para o exemplo 2 para a rede com 2 neurônios para cada classe.

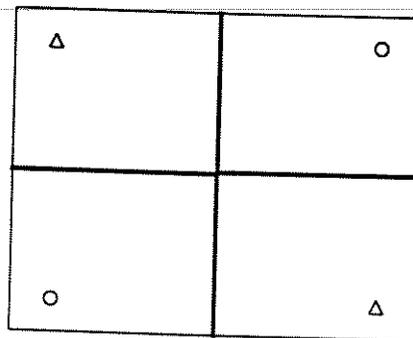


Fig. 2.32 - Superfície de separação para o exemplo 2: rede neural; norma N_2 . rede inicia com 2 neurônios por classe e rede final com 2 neurônios para a cada classe.

2.6.2.3 - Exemplo 3

Para o exemplo 3, além da norma N_2 , foram testadas as normas N_1 e N_∞ . Para os três casos a rede inicial continha 50 neurônios para cada classe. No final do treinamento este número foi reduzido bastante: 10 para classe 1 e 9 para a classe 2 para a norma N_1 ; 10 para classe 1 e 10 para a classe 2 para a norma N_2 ; 10 para classe 1 e 8 para a classe 2 para a norma N_∞ . Para permitir uma comparação dos resultados os pesos iniciais foram os mesmos para os três caso. Como pode ser visto os resultados são semelhantes. As superfícies de separação são compostas por segmentos de reta, figuras de 2.33 a 2.35.

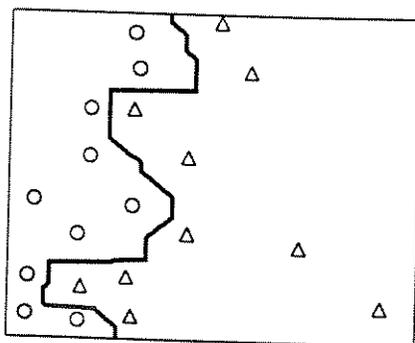


Fig. 2.33 - Superfície de separação para o exemplo 2: rede neural; norma N_1
 rede Inicial com 50 neurônios por classe e
 rede final 10 neurônios p/ classe 1 e 9 p/ classe 2.

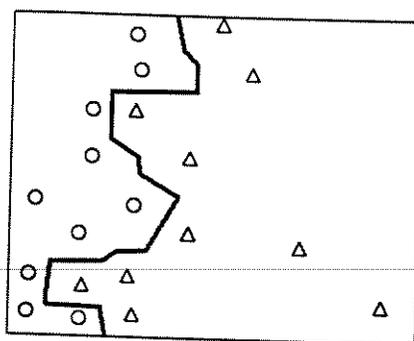


Fig. 2.34 - Superfície de separação para o exemplo 2: rede neural;
 norma N_2 rede Inicial com 50 neurônios por classe e
 rede final 10 neurônios p/ classe 1 e 10 p/ classe 2.

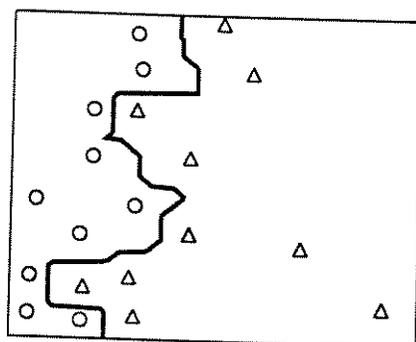


Fig. 2.35 - Superfície de separação para o exemplo 2: rede neural; norma N_∞ ;
 rede Inicial com 50 neurônios por classe e
 rede final 10 neurônios p/ classe 1 e 10 p/ classe 2.

2.6.2.4 - Exemplo 4

Para o exemplo 4, figura 2.36, a rede final contém 16 neurônios para a classe 1 e 8 para a classe 2, para uma configuração inicial com 50 neurônios por classe.

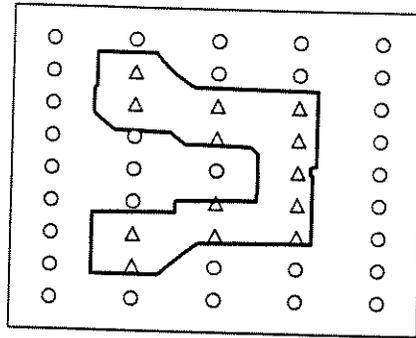


Fig. 2.36 - Superfície de separação para o exemplo 2:
rede neural; norma N_2 ;
rede Inicial com 50 neurônios por classe e
rede final 16 neurônios p/ classe 1 e 8 p/ classe 2.

Comparando os resultados obtidos para as duas topologias de rede, pode-se concluir que a superfície de separação é sempre linear quando é usada a rede neural com o algoritmo apresentado e que pode ser não-linear, dependendo da escolha das s e t-normas, para a rede *neurofuzzy and/or*.

2.7 - ANÁLISE DE COMPLEXIDADE DAS ESTRUTURAS DE REDE

Por simples inspeção, contando o número de operações e comparações efetuadas, verifica-se que o algoritmo de treinamento da rede *neurofuzzy and/or* é de ordem $O(2^n)$, no que diz respeito ao tempo de processamento. Com relação à capacidade de memória exigida para armazenamento dos pesos a complexidade é $O(N_p^n)$, para uma rede com mesmo número de subconjuntos para as entradas. Já a rede neural é de ordem $O(Nnc.n^2)$ para o tempo de processamento e capacidade de memória requerida. Isto mostra a inviabilidade de se utilizar a rede *neurofuzzy and/or* em problemas de classificação de padrões com número de coordenadas elevado. Por outro lado, o tempo para cálculo da rede *neurofuzzy and/or*, uma vez fixado o número de entradas, não aumenta com o aumento do número de neurônios por classe. Além disso, o tempo de treinamento é reduzindo, sendo que às vezes é possível classificar corretamente só com a inicialização dos pesos, conforme foi mostrado no exemplo 2. Isto não ocorre com a rede neural.

2.8 - CONCLUSÃO

Foram vistas neste capítulo duas estruturas de sistemas de classificação de padrões. A primeira estrutura é baseada numa rede *neurofuzzy and/or*, que utilizam algoritmo de treinamento competitivo, sem uso de derivadas. As vantagens desta rede são: geração automática da topologia, com extração de conhecimento diretamente da base de dados; capacidade de aprendizado; flexibilidade quanto a utilização de diversas **s** e **t**-normas; tempo de processamento constante e tempo de treinamento reduzido com aumento da dimensão da rede, fixando-se o número de coordenadas de entrada. A principal desvantagem é que o algoritmo é de complexidade exponencial com relação ao número de entradas da rede.

Deve-se também ressaltar que, para a rede convencional, auto-organizada, as normas N_1 e N_∞ apresentaram praticamente o mesmo resultado da norma N_2 , porém com tempo de processamento menor. A principal vantagem da rede neural com relação à rede *neurofuzzy and/or* é que o algoritmo é de complexidade polinomial. Por outro lado, se o número de entradas é fixado, o aumento do número de neurônios por classe implica em aumento de tempo de processamento, o que não ocorre para a rede *neurofuzzy and/or*.

Os resultados apresentados mostram que as estruturas apresentadas são alternativas viáveis para classificação de padrões.

CAPÍTULO 3

MODELAGEM DE FALHAS

3.1 - INTRODUÇÃO

Neste capítulo são desenvolvidos os modelos de três sistemas dinâmicos considerando os efeitos das falhas. O objetivo de tal desenvolvimento é ter uma ferramenta de projeto e teste dos sistemas de detecção e diagnóstico de falhas que serão estudados adiante.

São modelados três sistemas dinâmicos, a saber: sistema de acionamento elétrico utilizando máquina de corrente contínua (C.C.); sistema de acionamento elétrico utilizando máquina de corrente alternada (C.A.) e sistema de tanques interativos. Os modelos destes sistemas são não lineares e de ordem 3, 7 e 2, respectivamente. Para o sistema de acionamento C.C., além do modelo contínuo é desenvolvido o modelo discreto, que será usado no projeto dos observadores de estados em modos deslizantes. São apresentados resultados de simulações de algumas falhas modeladas.

3.2 - MODELAGEM DE FALHAS PARA O SISTEMA DE ACIONAMENTO C.C.

3.2.1 - MODELO DO SISTEMA DE ACIONAMENTO C.C.

O sistema de acionamento C.C. é composto por duas fontes de alimentação, conversores estáticos controlados, uma máquina de corrente contínua e uma carga mecânica, figura 3.1. O diagrama de blocos do sistema completo, incluindo o controle, é mostrado na figura 3.2. O controlador de velocidade, tipo proporcional integral, fornece o valor da corrente de armadura de referência. O controle da corrente da armadura é feito através da variação da tensão de alimentação. Esta tensão é função do ângulo de disparo do conversor, fornecido pelo controlador de corrente. A malha de campo, além de proporcionar o controle da corrente, permite que o sistema de acionamento opere à velocidade acima da nominal com potência constante, via enfraquecimento de campo (Bose, 1986).

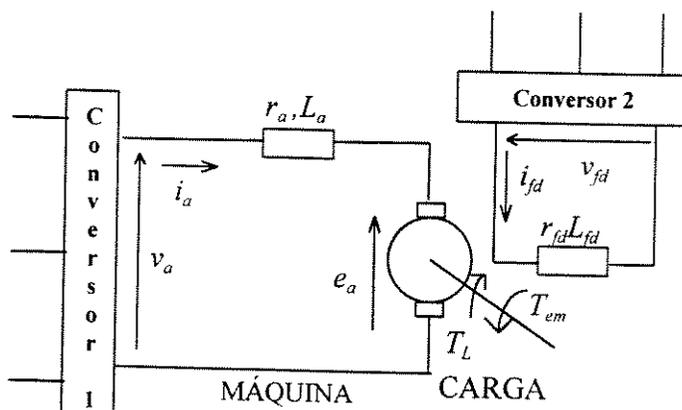


Fig. 3.1 - Representação do sistema de acionamento C.C.

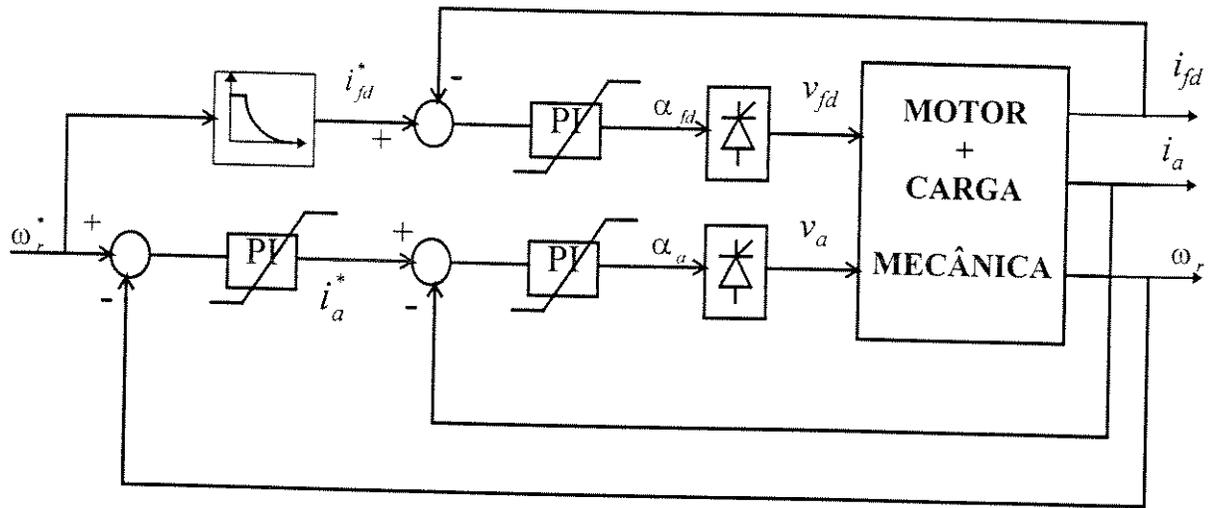


Fig. 3.2 - Diagrama de blocos do sistema de acionamento C.C.

onde

- v_a : tensão de alimentação do circuito da armadura;
- v_{fd} : tensão de alimentação do circuito de campo;
- i_a : corrente do circuito de armadura;
- i_{fd} : corrente do circuito de campo;
- ω_r : velocidade mecânica de rotação em rad/s;
- α_{fd} : ângulo de disparo do conversor do circuito de campo;
- α_a : ângulo de disparo do conversor do circuito de armadura;

sendo que as variáveis indexadas por n são referentes a valores nominais da máquina.

A corrente de campo de referência é calculada da seguinte forma:

$$i_{fd}^* = \begin{cases} i_{fd}^n, & \text{para } \omega_r^* \leq \omega_r^n \\ i_{fd}^n \cdot \frac{\omega_r^n}{\omega_r^*}, & \text{para } \omega_r^* > \omega_r^n \end{cases} \quad (3.1)$$

3.2.1.1 - Modelo da Máquina de Corrente Contínua

A máquina de corrente contínua possui o campo (excitação) independente da armadura, como mostrado na figura 3.1. Considerando as correntes i_a e i_{fd} e a velocidade ω_r como variáveis de estado, o modelo dinâmico para a máquina funcionando como motor é representado por:

$$\begin{cases} \frac{di_a}{dt} = \frac{1}{L_a} (v_a - r_a \cdot i_a - e_a) \\ \frac{di_{fd}}{dt} = \frac{1}{L_{fd}} (v_{fd} - r_{fd} \cdot i_{fd}) \\ \frac{d\omega_r}{dt} = \frac{1}{J_m} (T_{em} - B_m \cdot \omega_r - T_L) \end{cases} \quad (3.2)$$

sendo

$$\begin{cases} e_a = L_{afd} \cdot i_{fd} \cdot \omega_r \\ T_{em} = L_{afd} \cdot i_{fd} \cdot i_a \end{cases} \quad (3.3)$$

onde r_a : resistência do circuito da armadura;
 r_{fd} : resistência do circuito de campo;
 L_a : indutância do circuito da armadura;
 L_{fd} : indutância do circuito de campo;
 L_{afd} : indutância mútua armadura/campo;
 e_a : força-contratromotriz da armadura;
 T_{em} : conjugado eletromagnético;
 T_L : conjugado mecânico exigido pela carga;
 B_m : coeficiente de atrito viscoso;
 J_m : momento de inércia do sistema motor/carga.

Considerando que i_a , i_{fd} e ω_r são as variáveis medidas, colocando na forma de equação de estado, tem-se o modelo contínuo para a máquina de corrente contínua, representado por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{r_a}{L_a} & -\frac{L_{afd}}{L_a} \cdot x_3 & 0 \\ 0 & -\frac{r_{fd}}{L_{fd}} & 0 \\ \frac{L_{afd}}{J_m} \cdot x_2 & 0 & -\frac{B_m}{J_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{1}{L_{fd}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a \\ v_{fd} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \cdot [T_L] \quad (3.4)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \text{onde} \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} i_a \\ i_{fd} \\ \omega_r \end{bmatrix}$$

Nota-se que se trata de um sistema dinâmico não linear, resultante do produto de variáveis de estado.

Um modelo discreto para o sistema acima é representado pela equação (3.5). Para obtê-lo foi usada a mesma metodologia adotada por Silvino *et alli* (1993).

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} a_1 & a_2(k) & 0 \\ 0 & a_3 & 0 \\ a_4(k) & 0 & a_5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} [T_L(k)] \quad (3.5)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

onde

$$\left\{ \begin{array}{l} a_1 = a_1(r_a, L_a) = e^{-\frac{r_a}{L_a} h} \quad a_3 = a_3(r_{fd}, L_{fd}) = e^{-\frac{r_{fd}}{L_{fd}} h} \\ a_5 = a_5(B_m, J_m) = e^{-\frac{B_m}{J_m} h} \quad a_4(k) = a_4(B_m, J_m, x_2(k)) = \frac{(1-a_5)}{B_m} \cdot x_2(k) \\ b_1 = b_1(r_a, L_a) = \frac{1-a_1}{r_a} \\ a_2(k) = a_2(r_a, L_a, r_{fd}, L_{fd}, x_3(k)) = \\ = \frac{1}{r_{fd} \cdot L_a - r_a \cdot L_{fd}} [L_{afd} \cdot L_{fd} \cdot (a_3 - a_1) \cdot x_3(k) + (r_a \cdot L_{fd} \cdot a_1 - r_{fd} \cdot L_a \cdot a_3)] \\ b_2 = b_2(r_{fd}, L_{fd}) = \frac{1-a_3}{r_{fd}} \quad d_1 = d_1(B_m, J_m) = -\frac{1-a_5}{B_m} \end{array} \right. \quad (3.6)$$

3.2.1.2 - Modelo da Carga Mecânica

O modelo da carga mecânica é estabelecido pela equação de conjugado. Considerou-se uma carga genérica, cujo conjugado é:

$$T_L = C_0 + C_1 \cdot \omega_r + C_2 \cdot \omega_r^2 \quad (3.7)$$

3.2.1.3 - Modelo dos Conversores Estáticos

Os dois conversores, da armadura e do circuito de campo, são tipo Ponte de Graetz a Tiristor. É adotado o modelo estático para calcular a tensão de saída em função do ângulo de disparo (Bose, 1996):

$$\begin{aligned} v_a &= 2.34V_{oa} \cos \alpha_a \\ v_{fd} &= 2.34V_{ofd} \cos \alpha_{fd} \end{aligned} \quad (3.8)$$

sendo V_{oa} e V_{ofd} os valores eficazes das tensões nas entradas dos conversores da armadura e campo, respectivamente.

3.2.2 - MODELAGEM DAS FALHAS PARA O SISTEMA DE ACIONAMENTO C.C.

Do ponto de vista de classificação de falhas, o sistema de acionamento C.C. pode ser dividido em três grandes grupos: atuadores (conversores da armadura e do campo); planta ou processo (máquina e carga mecânica) e sensores (medidores de corrente e velocidade). Serão abordadas falhas relacionadas a estes três grupos.

3.2.2.1 - Falhas em Atuadores

O número de falhas que podem ocorrer nos conversores estáticos é muito grande, como, por exemplo, queima de componentes nos circuitos de potência e comando. Mas, do ponto de vista da máquina, as duas falhas mais graves são curto-circuito e desconexão do conversor. Estas falhas podem ser modeladas pelas chaves ideais CH_{aa} , CH_{afd} , CH_{cca} e CH_{ccfd} , figura 3.3.

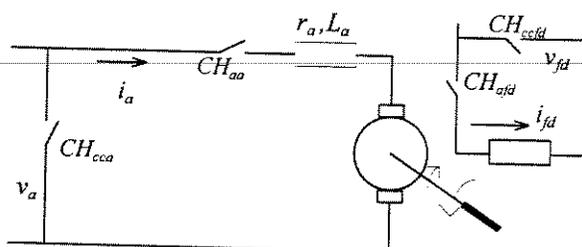


Fig. 3.3 - Representação do sistema de acionamento C.C. para falhas nos atuadores.

Associando os parâmetros:

- $k_{aa} \in \{0,1\}$ $\{ 1 \}$ indica a desconexão do conversor da armadura;
- $k_{afd} \in \{0,1\}$ $\{ 1 \}$ indica a desconexão do conversor do campo;
- $k_{cca} \in \{0,1\}$ $\{ 1 \}$ indica curto-circuito do conversor da armadura;
- $k_{ccfd} \in \{0,1\}$ $\{ 1 \}$ indica curto-circuito do conversor do campo;

aos estados das chaves, tal que:

- $k_i=0$ se CH_i estiver aberta e
- $k_i=1$ se CH_i estiver fechada, $i=aa, afd, cca$ e $ccfd$.

Considerando que uma falta de alimentação é caracterizada por uma corrente e um curto-circuito, por uma tensão nula, tem-se o modelo para falhas no circuito de alimentação da máquina dado por (3.9).

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} k_{aa} \cdot a_1 & k_{aa} \cdot a_2(k) & 0 \\ 0 & k_{afd} \cdot a_3 & 0 \\ a_4(k) & 0 & a_5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_{aa} \cdot \bar{k}_{cca} \cdot v_a(k) \\ k_{afd} \cdot \bar{k}_{ccfd} \cdot v_{fd}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \cdot [T_L(k)]$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \text{ onde } \bar{k}_i = 1 - k_i \quad (3.9)$$

3.2.2.2 - Falhas em Componentes da Planta

As falhas no conjunto máquina/carga consideradas são: curto-circuito entre espiras na bobina da armadura; curto-circuito entre espiras na bobina do campo; falha no sistema de ventilação e falha de lubrificação. As duas primeiras podem ser modeladas por variações nas resistências e indutâncias da armadura e do campo. Estas variações são modeladas pelos parâmetros:

- $k_{ca}^{r_a} \in [0,1]$ indica a variação na resistência r_a quando ocorre curto-circuito na bobina da armadura;
- $k_{ca}^{L_a} \in [0,1]$ indica a variação na indutância L_a quando ocorre curto-circuito na bobina da armadura;
- $k_{cfd}^{r_{fd}} \in [0,1]$ indica a variação na resistência r_{fd} quando ocorre curto-circuito na bobina do campo;
- $k_{cfd}^{L_{fd}} \in [0,1]$ indica a variação na indutância L_{fd} quando ocorre curto-circuito na bobina do campo.

A falha no sistema de ventilação se traduz em variações das resistências da armadura e do campo, provocadas pela elevação de temperatura da máquina. Já uma má lubrificação dos mancais, falha de lubrificação, pode ser modelada por uma variação do coeficiente de atrito.

Os parâmetros que representam estas duas falhas são:

- $k_{fv}^{r_a}$ indica a variação na resistência r_a quando ocorre falha no sistema de ventilação;
- $k_{fv}^{r_{fd}}$ indica a variação na resistência r_{fd} quando ocorre falha no sistema de ventilação;
- k_{fl} indica a variação no coeficiente de atrito, B_m , quando ocorre falha de lubrificação dos mancais.

O modelo que incorpora somente as falhas em componentes da planta é composto pelas equações de (3.5) a (3.8), onde, no cálculo dos coeficientes de (3.6), a_1 , $a_2(k)$, a_3 , $a_4(k)$, b_1 , b_2 e d_1 , os parâmetros nominais são substituídos por (3.10).

$$\begin{cases} r_a^f = k_{ca}^r \cdot k_{fv}^r \cdot r_a \\ L_a^f = k_{ca}^l \cdot k_{fv}^l \cdot L_a \\ r_{fd}^f = k_{cfd}^r \cdot k_{fv}^r \cdot r_{fd} \\ L_{fd}^f = k_{cfd}^l \cdot k_{fv}^l \cdot L_{fd} \\ B_m^f = k_{\beta} \cdot B_m \end{cases} \quad (3.10)$$

3.2.2.3 - Falhas de Instrumentação

Aqui são consideradas falhas bruscas (desligamentos) dos sensores de corrente (armadura e campo) e de velocidade, modeladas pelos parâmetros:

$k_{i_a}^f \in \{0,1\}$ { 1 } indica falha no sensor da corrente da armadura;

$k_{i_{fd}}^f \in \{0,1\}$ { 1 } indica falha no sensor da corrente do campo;

$k_{\omega_r}^f \in \{0,1\}$ { 1 } indica falha no sensor de velocidade da máquina.

Com relação ao modelo das falhas nos sensores, este difere do modelo representado pelas equações (3.5) a (3.8) somente com relação à equação de saída, que passa a ser calculada por:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} \bar{k}_{i_a}^f & 0 & 0 \\ 0 & \bar{k}_{i_{fd}}^f & 0 \\ 0 & 0 & \bar{k}_{\omega_r}^f \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \quad \text{onde } \bar{k}_i = 1 - k_i \quad (3.11)$$

3.2.3 - Modelo Geral das Falhas no Sistema de Acionamento C.C.

Considerando os dois tipos de falhas abordados anteriormente, tabela 1, tem-se o modelo completo dado por (3.12), (3.6), (3.7) e (3.8). O circuito representativo deste modelo é mostrado na figura 3.3.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} k_{aa} \cdot a_1^f & k_{aa} \cdot a_2^f(k) & 0 \\ 0 & k_{af} \cdot a_3^f & 0 \\ a_4^f(k) & 0 & a_5^f \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1^f & 0 \\ 0 & b_2^f \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_{aa} \cdot \bar{k}_{cca} \cdot v_a(k) \\ k_{af} \cdot \bar{k}_{ccfd} \cdot v_{fd}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_1^f \end{bmatrix} [T_L(k)]$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{k}_{i_a}^f & 0 & 0 \\ 0 & \bar{k}_{i_{fd}}^f & 0 \\ 0 & 0 & \bar{k}_{\omega_r}^f \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \text{ onde } \bar{k}_i = 1 - k_i \quad (3.12)$$

sendo que os coeficientes indexados pelo sobrescrito “f” são calculados substituindo os parâmetros nominais da máquina por (3.10).

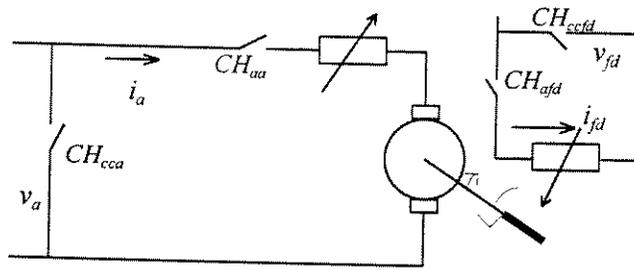


Fig. 3.4 - Representação do sistema de acionamento C.C. para os diversos tipos de falhas.

Tabela 1: Resumo das falhas no sistema de acionamento C.C.

Tipo de Falha	Variáveis Afetadas	Parâmetros Indicadores da Falha	Definição dos Parâmetros
abertura da bobina da armadura	$i_a=0$	k_{aa}	$\{0, 1\}$
abertura da bobina de campo	$i_{fd}=0$	k_{af}	$\{0, 1\}$
curto-circuito do conversor da armadura	$v_a=0$	k_{cFa}	$\{0, 1\}$
curto-circuito do conversor do campo	$v_{fd}=0$	k_{cFd}	$\{0, 1\}$
curto-circuito entre espiras da bobina de armadura	r_a e L_a	$k_{ca}^{r_a}$ e $k_{ca}^{L_a}$	$[0, 1]$
curto-circuito entre espiras da bobina de campo	r_{fd} e L_{fd}	$k_{cf}^{r_{fd}}$ e $k_{cf}^{L_{fd}}$	$[0, 1]$
falha no sistema de ventilação	r_a e r_{fd}	$k_{fv}^{r_a}$ e $k_{fv}^{r_{fd}}$	$[1, \infty)$
falha de lubrificação dos mancais	B_m	k_{fl}	$[0, \infty)$
falha no sensor de corrente da armadura	i_a	$k_{i_a}^f$	
falha no sensor de corrente da fonte	i_{fd}	$k_{i_{fd}}^f$	$\{0, 1\}$
falha no sensor de velocidade	ω_r	$k_{\omega_r}^f$	$\{0, 1\}$

3.2.4 - RESULTADOS DE SIMULAÇÕES DO SISTEMA DE ACIONAMENTO C.C.

3.2.4.1 - Avaliação do Modelo Discreto

As figuras abaixo mostram a simulação da partida do sistema de acionamento, cujos parâmetros são apresentados no Apêndice A. A máquina parte a vazio e em $t=1.8$ segundo é aplicada uma carga constante no eixo. As figuras 3.5a, 3.5b e 3.5c mostram curvas das correntes e tensões do campo e da armadura, velocidade e conjugado de carga. Todas as grandezas são normalizadas, tendo como bases valores nominais da máquina. A condição de operação é com enfraquecimento de campo, figura 3.5, pois a velocidade de referência é o dobro da nominal. A simulação foi feita utilizando os dois modelos: discreto e contínuo. Para o modelo contínuo as equações foram resolvidas pelo método de Runge-Kutta de quarta ordem, com um intervalo de integração de $10 \mu\text{s}$. O período de amostragem para calcular o modelo discreto foi de 2 ms. As curvas de erros, figura 3.6, mostram que o modelo discreto é uma boa aproximação do contínuo, apresentando um erro máximo em torno de 10% durante um curto período transitório, caindo praticamente a zero em regime permanente. Os erros mostrados na figura 3.6 são calculados da seguinte forma:

$$e_{x_i} = \frac{x_i^c - x_i^d}{x_i^{\text{nominal}}}$$

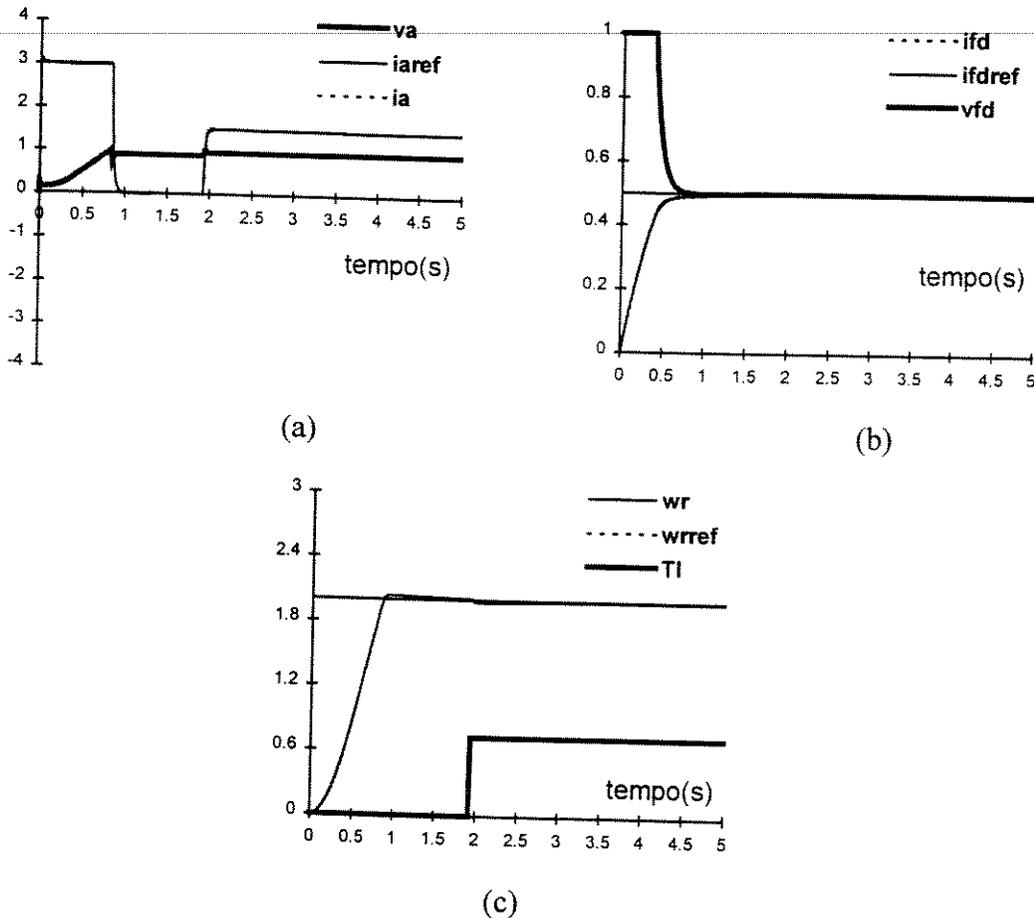


Fig. 3.5 - Simulação da partida da máquina de corrente contínua

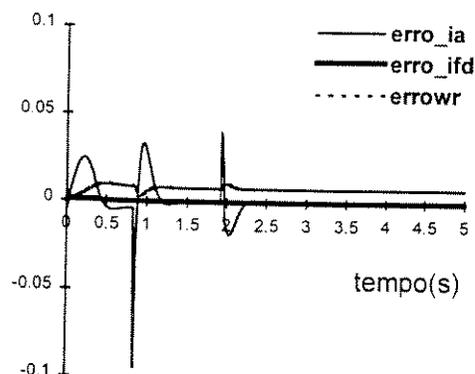


Fig. 3.6 - Diferença entres os modelos discretos e contínuos

2.2.4.2 - Simulações de Falhas

As figuras 3.7, 3.8 e 3.9 mostram as curvas de correntes de campo e de armadura e velocidade para a partida da máquina com ocorrência de falhas. As falhas simuladas foram: abertura da armadura; abertura do campo e falha no sensor de velocidade. Uma desconexão do conversor da armadura provoca a parada da máquina, pois, com a corrente de armadura nula, o conjugado eletromagnético também se anula. Já a corrente de campo não é afetada, porque o circuito de campo é independente, figura 3.7. A desconexão do conversor do campo provoca a parada da máquina e, com isto, um aumento da corrente de armadura, que fica limitada somente pela resistência r_a , porque a força-contratromotriz se anula com a velocidade, conforme expressão (3.3). Quando ocorre uma falha no sensor de velocidade, o controlador fornece uma ação de controle, solicitando uma corrente de armadura de referência maior, para efetuar a correção do erro. No instante da ocorrência da falha o sistema estava funcionando em condição nominal; isto impede que a corrente de armadura acompanhe o aumento da referência, pois a tensão de armadura não pode ser aumentada e nem ocorreu redução de força-contratromotriz.

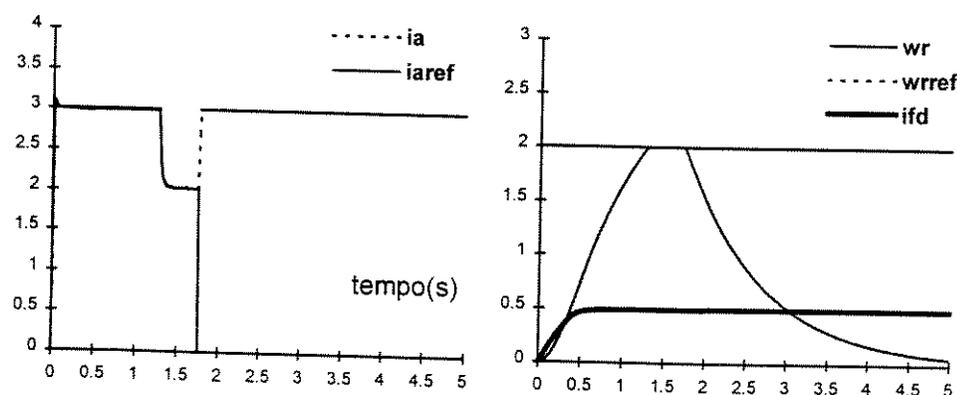


Fig. 3.7 - Simulação da desconexão do conversor da armadura

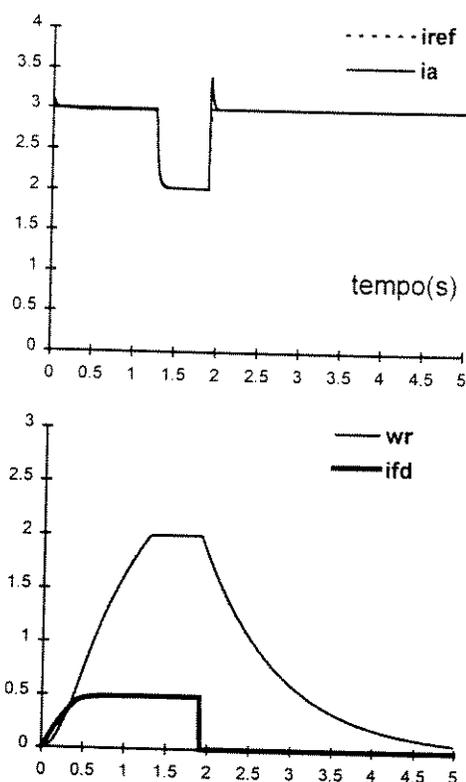


Fig. 3.8 - Simulação da desconexão do conversor de campo

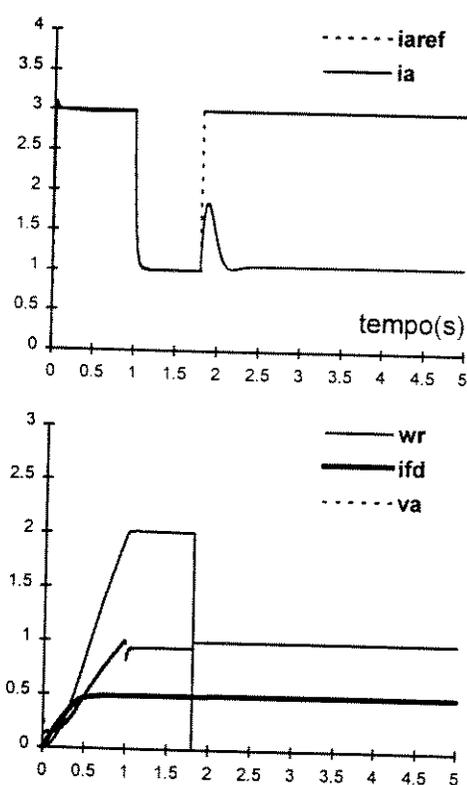


Fig. 3.9 - Simulação de falha no sensor de velocidade

3.3 - MODELAGEM DE FALHAS PARA SISTEMA DE ACIONAMENTO C.A.

3.3.1 - MODELAGEM DO SISTEMA ACIONAMENTO C.A.

O sistema de acionamento C.A., figura 3.10, é composto por uma fonte de alimentação (conversor estático ou rede elétrica), um motor de indução trifásico tipo rotor em gaiola e uma carga mecânica.

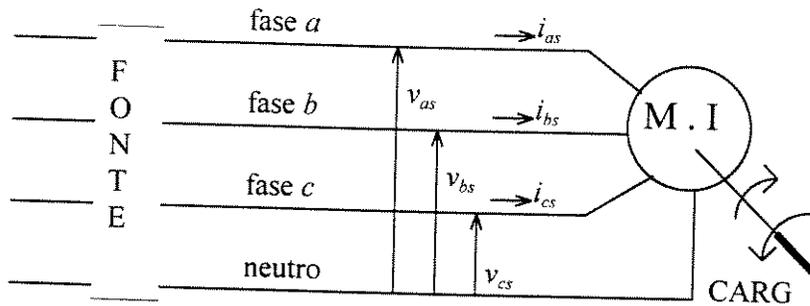


Fig. 3.10 - Sistema de acionamento C.A.

3.3.1.1 - Modelo da Máquina de Indução

O modelo da máquina de indução é de 7a. ordem, não linear e com parâmetros variantes no tempo. Considerando os enlaces de fluxo e velocidade do rotor como variáveis de estados, para um sistema de eixos girando à uma velocidade arbitrária ω , o modelo da máquina (Krause, 1986) é dado por:

$$\begin{bmatrix} \dot{\lambda}_{qs} \\ \dot{\lambda}_{ds} \\ \dot{\lambda}_{0s} \\ \dot{\lambda}_{qr} \\ \dot{\lambda}_{dr} \\ \dot{\lambda}_{0r} \\ \dot{\omega}_r \end{bmatrix} = \begin{bmatrix} -a_1 & -\omega & 0 & a_2 & 0 & 0 & 0 \\ \omega & -a_1 & 0 & 0 & 0 & a_2 & 0 \\ 0 & 0 & -a_3 & 0 & 0 & 0 & 0 \\ a_4 & 0 & 0 & -a_5 & -(\omega - \omega_r) & 0 & 0 \\ 0 & a_4 & 0 & (\omega - \omega_r) & -a_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_6 & 0 \\ a_7 \cdot \lambda_{dr} & -a_7 \cdot \lambda_{qr} & 0 & 0 & 0 & 0 & a_8 \end{bmatrix} \begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda_{qr} \\ \lambda_{dr} \\ \lambda_{0r} \\ \omega_r \end{bmatrix} + \begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{0s} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -T_L \end{bmatrix} \quad (3.13)$$

Uma variável G qualquer (tensão, corrente ou enlace de fluxo) em componente de $dq0$ é obtida a partir das componentes naturais (abc) utilizando a matriz de transformação $[K_s]$:

$$\begin{bmatrix} G_{dq0} \end{bmatrix} = [K_s] \cdot [G_{abc}]$$

$$\begin{bmatrix} G_q \\ G_d \\ G_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ \sin(\theta) & \sin(\theta - 2\pi/3) & \sin(\theta + 2\pi/3) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} G_a \\ G_b \\ G_c \end{bmatrix} \quad \text{onde } \theta = \theta_0 + \int_0^t \omega \cdot d\tau \quad (3.14)$$

sendo

$$\begin{cases} a_0 = L_{ss} \cdot L_{rr} - M^2 & a_1 = \frac{L_{rr}}{a_0} \cdot R_s & a_2 = \frac{M}{a_0} \cdot R_s \\ a_3 = \frac{1}{L_{ls}} \cdot R_s & a_4 = \frac{M}{a_0} \cdot R_r & a_5 = \frac{L_{ss}}{a_0} \cdot R_r \\ a_6 = \frac{1}{L_{lr}} \cdot R_r & a_7 = \frac{3}{2} \cdot \frac{P}{2} \cdot \frac{M}{a_0 \cdot J_M} & a_8 = \frac{B_M}{J_M} \end{cases} \quad (3.15)$$

onde:

- R_s : resistência do estator;
- R_r : resistência do rotor;
- L_{ss} : indutância própria do estator;
- L_{rr} : indutância própria do rotor;
- L_{ls} : indutância de dispersão do estator;
- L_{lr} : indutância de dispersão do rotor;
- M : indutância mútua entre o estator e rotor;
- P : número de pólos da máquina;
- J_m : momento de inércia;
- B_m : coeficiente de atrito viscoso;
- T_L : conjugado de carga;
- i : corrente instantânea;
- v : tensão instantânea;
- λ : enlace de fluxo;
- ω : velocidade angular.

3.3.1.2 - Modelo da Carga Mecânica

O modelo da carga mecânica é o mesmo do sistema de acionamento C.C., ou seja, o conjugado de carga é dado por:

$$T_L = C_0 + C_1 \cdot \omega_r + C_2 \cdot \omega_r^2 \quad (3.16)$$

3.3.1.3 - Modelo da Fonte de Alimentação

A fonte usada foi a rede elétrica, cujo modelo é dado por:

$$\begin{cases} v_{Fa} = V_m \cdot \cos(2\pi f \cdot t) \\ v_{Fb} = V_m \cdot \cos(2\pi f \cdot t - 2\pi / 3) \\ v_{Fc} = V_m \cdot \cos(2\pi f \cdot t + 2\pi / 3) \end{cases} \quad (3.17)$$

sendo f a frequência elétrica da rede.

Para operação normal, e desprezando a impedância do circuito de alimentação, as tensões nos terminais da máquina são iguais às da fonte.

3.3.2 - MODELAGEM DE FALHAS PARA O SISTEMA DE ACIONAMENTO C.A.

Para o sistema de acionamento C.A., foram consideradas somente falhas relacionadas com o circuito de alimentação da máquina, a saber: curto-circuito da fase/neutro e abertura de fase (falta de alimentação).

3.3.2.1 - Modelagem do Curto-Circuito

De forma análoga ao desenvolvimento feito para o acionamento C.C., esta falha é modelada considerando os seguintes parâmetros:

- $k_{ca} \in \{0,1\}$ indica curto-circuito da fase "a" para o neutro;
- $k_{cb} \in \{0,1\}$ indica curto-circuito da fase "b" para o neutro;
- $k_{cc} \in \{0,1\}$ indica curto-circuito da fase "c" para o neutro.

Sendo estes parâmetros associados aos estados das chaves CH_{ca} , CH_{cb} , e CH_{cc} , respectivamente, figura 3.11.

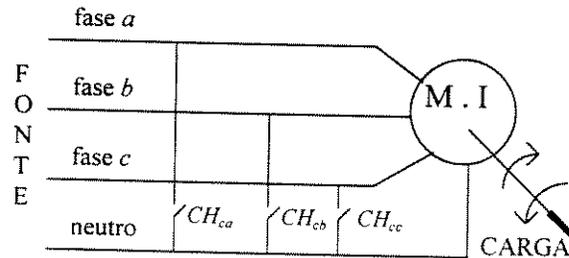


Fig. 3.11 - Diagrama do sistema de acionamento C.A. para modelagem de falhas de curto-circuito.

Sendo que:

$$\begin{aligned} k_{ci} &= 0 \text{ se } CH_{ci} \text{ estiver aberta e} \\ k_{ci} &= 1 \text{ se } CH_{ci} \text{ estiver fechada, para } i=a, b, c. \end{aligned}$$

O curto-circuito da fase "i" para o neutro é modelado fazendo o valor da tensão dessa fase para o neutro assumir o valor zero, ou seja:

$$\begin{cases} v_{as} = \bar{k}_{ca} \cdot v_{Fa} \\ v_{bs} = \bar{k}_{cb} \cdot v_{Fb} \\ v_{cs} = \bar{k}_{cc} \cdot v_{Fc} \end{cases} \quad \text{onde } \bar{k}_{ci} = 1 - k_{ci}, \quad i = a, b, c \quad (3.18)$$

3.3.2.2 - Modelagem da abertura de fase

A modelagem da abertura de fase não é uma tarefa tão simples como a de curto-circuito. Isto porque durante a abertura de fase não se sabe qual é o valor da tensão na fase aberta, chamada de tensão de flutuação (Cruz *et alii*, 1990). Isto quer dizer que a tensão, que era um dado de entrada, passa ser uma incógnita. Por outro lado, sabe-se que, para a fase aberta, a corrente e sua derivada são nulas. Portanto, com esta informação é possível calcular o valor da tensão de flutuação, para que seja possível resolver as equações de estado do modelo da máquina. De modo análogo ao anterior, estas falhas são modeladas por chaves ideais, CH_{ai} , onde $i=a, b$ e c , figura 3.12.

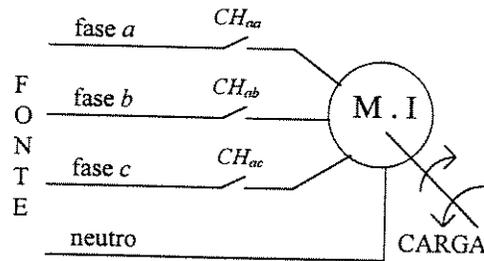


Fig. 3.12- Diagrama do sistema de acionamento C.A. para modelagem de falhas de abertura de fase.

O cálculo da tensão de flutuação é feita a partir da derivada da corrente, que é nula para esta fase. É adotada a abordagem proposta por Cruz *et alii* (1990). Para um sistema de eixos fixo no estator da máquina, tem-se a seguinte equação:

$$[A_1] \begin{bmatrix} pi_{as} \\ pi_{bs} \\ pi_{cs} \end{bmatrix} = [A_2] \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} + [A_3] \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \end{bmatrix} + [A_4] \begin{bmatrix} i_{qr} \\ i_{dr} \\ i_{0r} \end{bmatrix} \quad (3.19)$$

sendo que o operador p significa derivada em relação ao tempo.

A relação das correntes com os enlaces de fluxo é dada por:

$$\begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i_{qr} \\ i_{dr} \\ i_{0r} \end{bmatrix} = \begin{bmatrix} a_9 & 0 & 0 & -a_{10} & 0 & 0 \\ 0 & a_9 & 0 & 0 & -a_{10} & 0 \\ 0 & 0 & -a_{11} & 0 & 0 & 0 \\ -a_{10} & 0 & 0 & a_{12} & 0 & 0 \\ 0 & -a_{10} & 0 & 0 & a_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & -a_{13} \end{bmatrix} \begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda_{qr} \\ \lambda_{dr} \\ \lambda_{0r} \end{bmatrix} \quad (3.20)$$

onde:

$$[A_1] = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ 0 & -\sqrt{3} & \sqrt{3} \\ 1 & 1 & 1 \end{bmatrix} \quad (3.21)$$

$$[A_2] = \frac{1}{3} \begin{bmatrix} 2 \cdot a_9 & -a_9 & -a_9 \\ 0 & -\sqrt{3} \cdot a_9 & \sqrt{3} \cdot a_9 \\ a_{11} & a_{11} & a_{11} \end{bmatrix} \quad (3.22)$$

$$[A_3] = \begin{bmatrix} -a_1 & -a_{14} \cdot \omega_r & 0 \\ a_{14} \cdot \omega_r & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix} \quad (3.23)$$

$$[A_4] = \begin{bmatrix} -a_4 & -a_{15} \cdot \omega_r & 0 \\ a_{15} \cdot \omega_r & -a_4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.24)$$

$$\begin{cases} a_9 = \frac{L_{rr}}{a_0} & , & a_{10} = \frac{M}{a_0} & & a_{11} = \frac{1}{L_{ls}} \\ a_{12} = \frac{L_{ss}}{a_0} & & a_{13} = \frac{1}{L_{lr}} & & a_{14} = \frac{M^2}{a_0} \\ a_{15} = \frac{L_{rr} \cdot M}{a_0} & & & & \end{cases} \quad (3.25)$$

Generalizando a proposta de (Cruz *et alii*, 1990) pode-se escrever as equações que permitem simular as falhas de abertura de fases:

$$[A] \cdot [X] = [b] \quad (3.26)$$

sendo

$$[A] = \frac{1}{3} \begin{bmatrix} 2 \cdot (k_{aa} - a_9 \cdot \bar{k}_{aa}) & -(k_{ab} - a_9 \cdot \bar{k}_{ab}) & -(k_{ac} - a_9 \cdot \bar{k}_{ac}) \\ 0 & -\sqrt{3} \cdot (k_{ab} - a_9 \cdot \bar{k}_{ab}) & \sqrt{3} \cdot (k_{ac} - a_9 \cdot \bar{k}_{ac}) \\ (k_{aa} - a_9 \cdot \bar{k}_{aa}) & (k_{ab} - a_9 \cdot \bar{k}_{ab}) & (k_{ac} - a_9 \cdot \bar{k}_{ac}) \end{bmatrix} \quad (3.27)$$

$$[X] = \begin{bmatrix} k_{aa} \cdot p i_{as} + \bar{k}_{aa} \cdot v_{as} \\ k_{ab} \cdot p i_{bs} + \bar{k}_{ab} \cdot v_{bs} \\ k_{ac} \cdot p i_{cs} + \bar{k}_{ac} \cdot v_{cs} \end{bmatrix} \quad (3.28)$$

$$[b] = \frac{1}{3} \begin{bmatrix} 2 \cdot (a_9 \cdot k_{aa} - \bar{k}_{aa}) & -(a_9 \cdot k_{ab} - \bar{k}_{ab}) & -(a_9 \cdot k_{ac} - \bar{k}_{ac}) \\ 0 & -\sqrt{3} \cdot (a_9 \cdot k_{ab} - \bar{k}_{ab}) & \sqrt{3} \cdot (a_9 \cdot k_{ac} - \bar{k}_{ac}) \\ (a_9 \cdot k_{aa} - \bar{k}_{aa}) & (a_9 \cdot k_{ab} - \bar{k}_{ab}) & (a_9 \cdot k_{ac} - \bar{k}_{ac}) \end{bmatrix} \begin{bmatrix} \bar{k}_{aa} \cdot p i_{as} + k_{aa} \cdot v_{as} \\ \bar{k}_{ab} \cdot p i_{bs} + k_{ab} \cdot v_{bs} \\ \bar{k}_{ac} \cdot p i_{cs} + k_{ac} \cdot v_{cs} \end{bmatrix} + [A_3] \cdot \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \end{bmatrix} + [A_4] \cdot \begin{bmatrix} i_{qr} \\ i_{dr} \\ i_{0r} \end{bmatrix} \quad (3.29)$$

sendo $\bar{k}_{ai} = 1 - k_{ai}$, $i = a, b, c$ e

- $k_{aa} \in \{0,1\}$ indica abertura da fase “a”;
 $k_{ab} \in \{0,1\}$ indica abertura da fase “b”;
 $k_{ac} \in \{0,1\}$ indica abertura da fase “c”.

O sistema acima pode ser resolvido utilizando qualquer método numérico de resolução de sistemas lineares ou algebricamente. Como somente no caso de abertura das três fases é necessário encontrar solução para as três incógnitas, o método recomendado é o de Cramer. Neste caso, as equações para as tensões de flutuação são:

$$\begin{cases} v_{as}^{fa} = \frac{\Delta_1}{\Delta} \\ v_{bs}^{fb} = \frac{\Delta_2}{\Delta} \\ v_{cs}^{ca} = \frac{\Delta_3}{\Delta} \end{cases} \quad (3.30)$$

onde Δ é determinante de $[A]$ e Δ_i é determinante de $[A]$ onde a coluna “i” foi substituída pelo vetor $[b]$.

Para exemplificar considere as seguintes situações:

a) Abertura da Fase “a”

Para a abertura da fase, resolvendo algebricamente o sistema (3.26) obtém-se:

$$v_{as} = -\frac{3}{2 \cdot a_9 + a_{11}} \cdot (b_1 + b_3) \quad (3.31)$$

b) Abertura da Fase “b”

$$v_{bs} = \frac{3}{2 \cdot a_9 + a_{11}} \cdot \left(\frac{1}{2} \cdot b_1 + \frac{\sqrt{3}}{2} \cdot b_2 + b_3 \right) \quad (3.32)$$

c) Abertura da Fase “c”

$$v_{bs} = -\frac{3}{2 \cdot a_9 + a_{11}} \cdot \left(-\frac{1}{2} \cdot b_1 + \frac{\sqrt{3}}{2} \cdot b_2 + b_3 \right) \quad (3.33)$$

O mesmo procedimento pode ser adotado no caso de duas ou três fases abertas (Cruz *et alii*, 1990).

3.3.3 - MODELO GERAL DE FALHAS PARA O SISTEMA DE ACIONAMENTO C.A.

O modelo geral que permite simular as falhas no circuito de alimentação para o sistema de acionamento C.A. é composto pelo conjunto de equações (3.34) a (3.36). O diagrama do sistema considerando as falhas é mostrado na figura 3.13.

$$\begin{bmatrix} \dot{\lambda}_{qs} \\ \dot{\lambda}_{ds} \\ \dot{\lambda}_{0s} \\ \dot{\lambda}_{qr} \\ \dot{\lambda}_{dr} \\ \dot{\lambda}_{0r} \\ \dot{\omega}_r \end{bmatrix} = \begin{bmatrix} -a_1 & -\omega & 0 & a_2 & 0 & 0 & 0 \\ \omega & -a_1 & 0 & 0 & 0 & a_2 & 0 \\ 0 & 0 & -a_3 & 0 & 0 & 0 & 0 \\ a_4 & 0 & 0 & -a_5 & -(\omega - \omega_r) & 0 & 0 \\ 0 & a_4 & 0 & (\omega - \omega_r) & -a_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_6 & 0 \\ a_7 \cdot \lambda_{dr} & -a_7 \cdot \lambda_{qr} & 0 & 0 & 0 & 0 & a_8 \end{bmatrix} \begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda_{qr} \\ \lambda_{dr} \\ \lambda_{0r} \\ \omega_r \end{bmatrix} + \begin{bmatrix} v_{qs}^f \\ v_{ds}^f \\ v_{0s}^f \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -T_L \end{bmatrix} \quad (3.34)$$

$$\begin{bmatrix} v_{qs}^f \\ v_{ds}^f \\ v_{0s}^f \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ \sin(\theta) & \sin(\theta - 2\pi/3) & \sin(\theta + 2\pi/3) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} v_{as}^f \\ v_{bs}^f \\ v_{cs}^f \end{bmatrix} \quad (3.35)$$

$$\begin{bmatrix} v_{as}^f \\ v_{bs}^f \\ v_{cs}^f \end{bmatrix} = \begin{bmatrix} \bar{k}_{ca} \cdot k_{aa} \cdot v_{Fa} + k_{ca} \cdot \bar{k}_{aa} \cdot \frac{\Delta_1}{\Delta} \\ \bar{k}_{cb} \cdot k_{ab} \cdot v_{Fb} + k_{cb} \cdot \bar{k}_{ab} \cdot \frac{\Delta_2}{\Delta} \\ \bar{k}_{cc} \cdot k_{aa} \cdot v_{Fc} + k_{cc} \cdot \bar{k}_{ac} \cdot \frac{\Delta_3}{\Delta} \end{bmatrix} \quad (3.36)$$

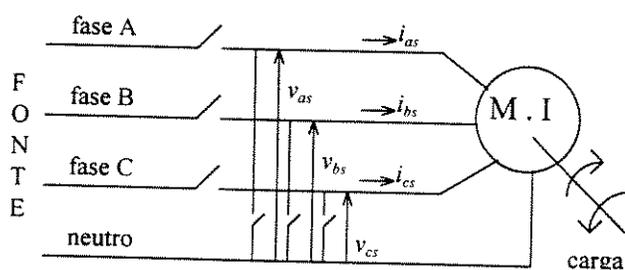


Fig. 3.13 - Diagrama do sistema de acionamento C.A. para modelagem das falhas

Com este modelo é possível simular diferentes combinações de falhas no sistema, conforme mostradas na tabela 2.

Tabela 2 - Tipo de falhas no sistema de acionamento C.A.

Tipo de Falha	Parâmetros Indicadores da Falha	Definição dos Parâmetros
curto-circuito entre a fase "a" e neutro	k_{ca}	{0,1}
curto-circuito entre a fase "b" e neutro	k_{cb}	{0,1}
curto-circuito entre a fase "c" e neutro	k_{cc}	{0,1}
curto-circuito entre as fases "a,b" e neutro	k_{ca} e k_{cb}	{0,1}
curto-circuito entre as fases "a,c" e neutro	k_{ca} e k_{cc}	{0,1}
curto-circuito entre as fases "b,c" e neutro	k_{cb} e k_{cc}	{0,1}
curto-circuito entre as fases "a,b,c" e neutro	k_{ca} ; k_{cb} e k_{cc}	{0,1}
fase "a" aberta	k_{aa}	{0,1}
fase "b" aberta	k_{ab}	{0,1}
fase "c" aberta	k_{ac}	{0,1}
fases "a,b" abertas	k_{aa} e k_{ab}	{0,1}
fases "a,c" abertas	k_{aa} e k_{ac}	{0,1}
fases "b,c" abertas	k_{ab} e k_{ac}	{0,1}
fases "a,b,c" abertas	k_{aa} ; k_{ab} e k_{ac}	{0,1}

3.3.3 - RESULTADOS DE SIMULAÇÕES DE FALHAS

As figuras de 3.14 a 3.17 mostram a partida da máquina e em $t=0,5$ segundos é provocada uma falha, curto-circuito da fase "a" para o neutro. Esta situação leva a tensão da fase a zero e um aumento das correntes em todas as fases. Devido ao desequilíbrio das tensões do estator, a componente de seqüência negativa proporciona uma redução do conjugado eletromagnético e, conseqüentemente, uma redução da velocidade, figura 3.17. As figuras de 3.18 a 3.21 mostram aberturas das fases "b" e "c". Para esta situação as correntes nas respectivas fases são nulas e as tensões são as de flutuação. Como o desequilíbrio das correntes é grande, a componente de seqüência negativa proporciona uma redução muito grande de conjugado eletromagnético, que fica menor que o conjugado de carga. Isto provoca a parada da máquina, figura 3.21.

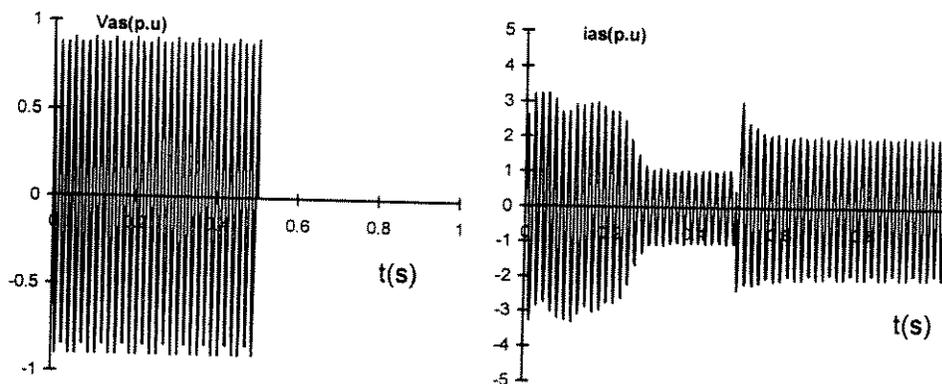


Fig. 3.14 - Tensão e corrente na fase "a" para curto-circuito na fase "a"

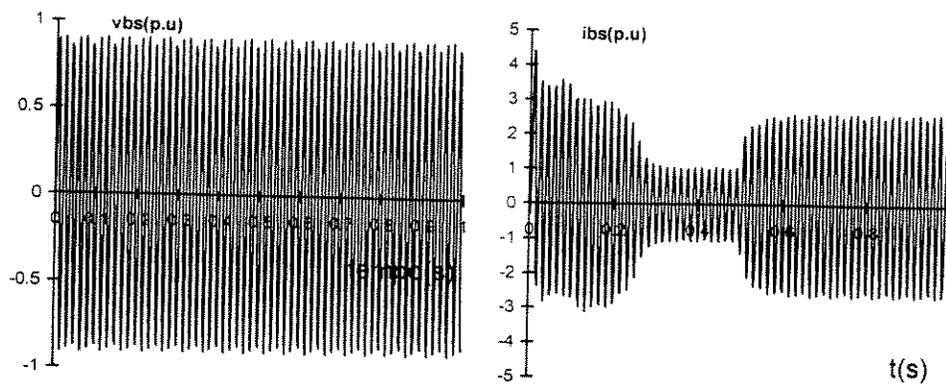


Fig. 3.15 - Tensão e corrente na fase "b" para curto-circuito na fase "a"

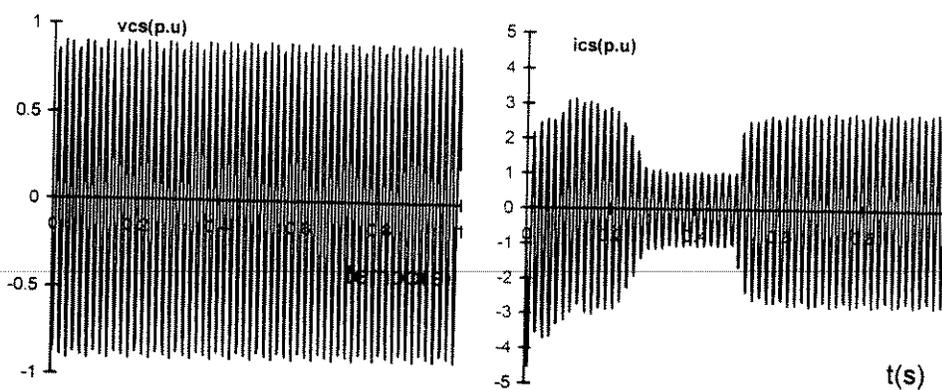


Fig. 3.16 - Tensão e corrente na fase "c" para curto-circuito na fase "a"

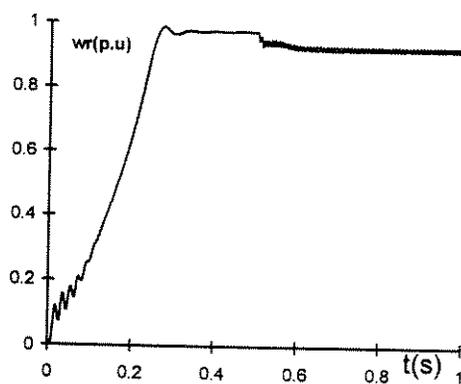


Fig. 3.17 - Velocidade de rotação para curto-circuito na fase "a"

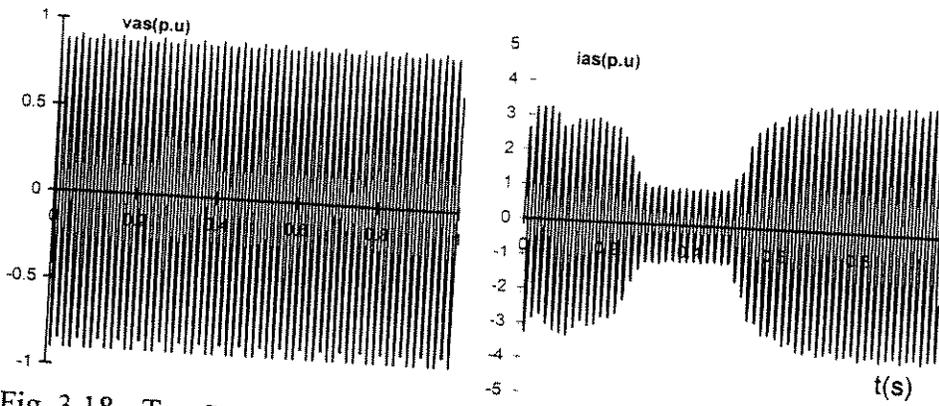


Fig. 3.18 - Tensão e corrente na fase "a" para abertura das fases "b" e "c"

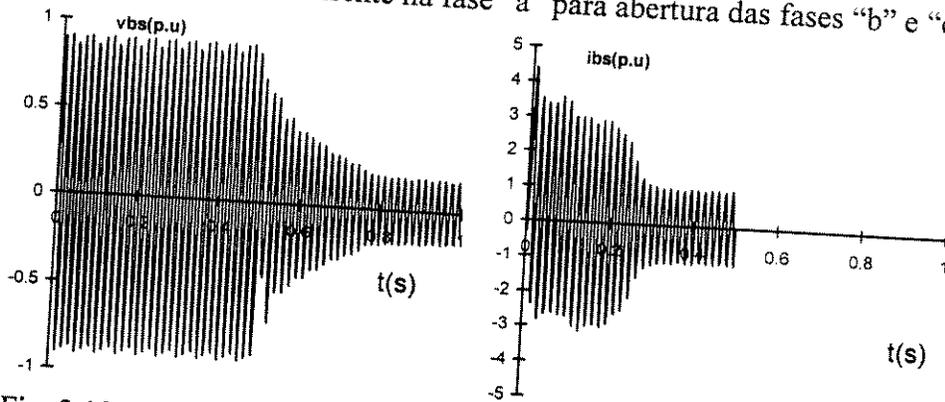


Fig. 3.19 - Tensão e corrente na fase "b" para abertura das fases "b" e "c"

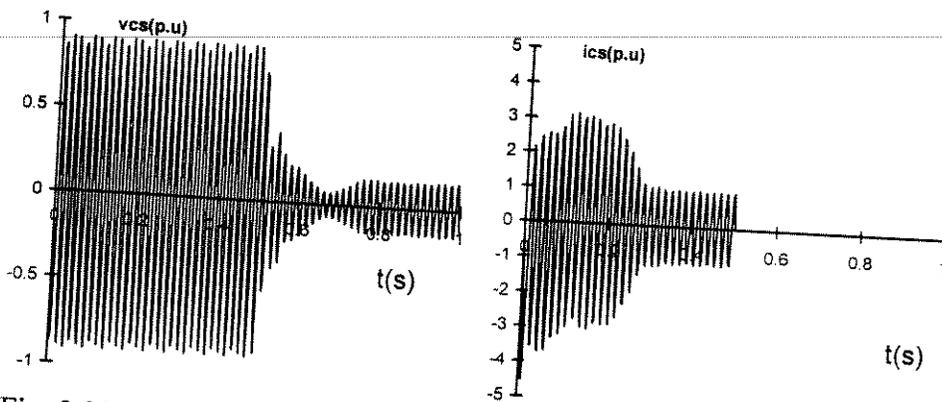


Fig. 3.20 - Tensão e corrente na fase "c" para abertura das fases "b" e "c"

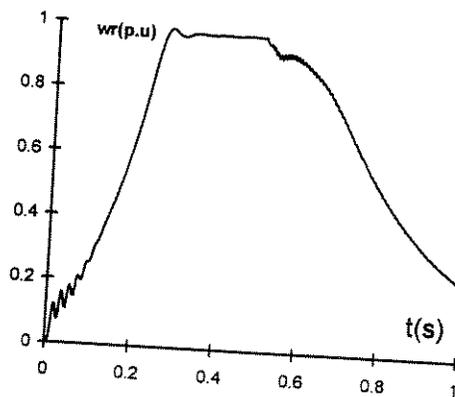


Fig. 3.21 - Velocidade de rotação para abertura das fases "b" e "c"

3.4 - MODELAGEM DE FALHAS PARA O SISTEMA DE TANQUES INTERATIVOS

A modelagem de falhas apresentada neste item foi desenvolvida e validada por Pereira (1996). Boa parte do texto apresentado neste item foi retirada do referido trabalho, sendo que as equações foram escritas de forma mais compacta.

3.4.1 - DESCRIÇÃO DO SISTEMA

A descrição detalhada do Sistema de Tanques Interativos (ou acoplados)-STI, do Laboratório de Controle de Processos Industriais do CPDEE-UFMG, está presente em (Jota *et alii*, 1995; Braga, 1994). A figura 3.22 mostra o diagrama simplificado do sistema. Um microcomputador PC386 executa o software de supervisão e controle, ligado a um Controlador Lógico Programável (PLC), através de uma porta serial RS-232. O PLC é encarregado da aquisição de dados das variáveis do processo, de atuar na planta de acordo com os dados fornecidos pelo microcomputador, além de executar o controle lógico do intertravamento necessário para o funcionamento correto do sistema. Entre o PLC e a planta há um painel de operação que permite, além da visualização das variáveis, o controle lógico e a operação manual do processo.

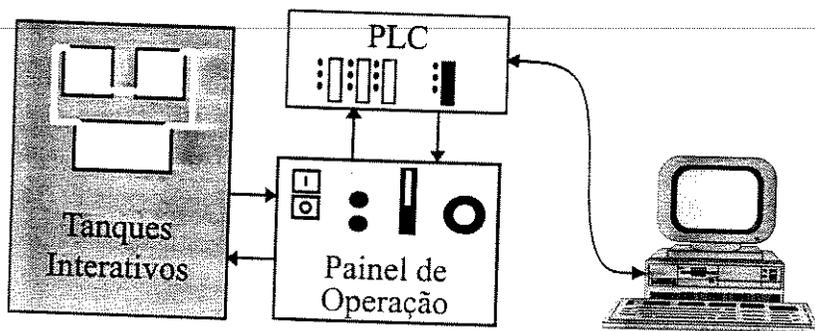


Fig. 3.22 - Diagrama de blocos do Sistema de Tanques Interativos (Braga, 1994)

A figura 3.23 mostra o diagrama do fluxo de massa nos Tanques Interativos. O objetivo é controlar o nível do tanque TQ-3, h_3 , e a sua vazão de saída, q_o . O ciclo de operação do sistema consiste no bombeamento da água do tanque TQ-1 para o tanque TQ-2 através da bomba BA-1. A água flui para o tanque TQ-3 através do princípio dos vasos comunicantes; a intensidade deste acoplamento é controlada através das válvulas FV-3 e FV-4 (a válvula FV-5 fica permanentemente fechada durante o funcionamento normal). O ciclo se completa com o bombeamento da água do TQ-3 de volta para o TQ-1 através da bomba BA-2. Com isto, o controle do nível no tanque TQ-3 é feito a partir da atuação da válvula pneumática FCV-1, que controla a vazão de entrada de TQ-2 e, conseqüentemente, controla a vazão de entrada de TQ-3. Já a vazão de saída de TQ-3 é controlada atuando-se na válvula pneumática FCV-2.

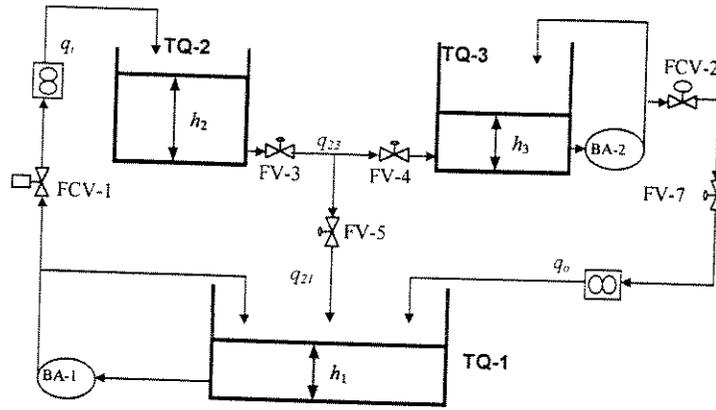


Fig. 3.23 - Fluxo de massa nos Tanques Interativos

3.4.2 - MODELO DO SISTEMA DE TANQUES INTERATIVOS

O modelo matemático do sistema de tanques interativos é obtido a partir do balanço de massa entre os tanques TQ-2 e TQ-3 e utilizando a Lei de Bernouli (Braga, 1994). Tomando-se: $x_1 = h_2$, $x_2 = h_3$, $y_1 = h_3$, $y_2 = q_o$ e $y_3 = q_i$, o modelo do sistema é dado por:

$$\begin{cases} \dot{x}_1 = \frac{1}{A} [q_i(R_1, u_1, p_1) - q_{23}(x_1, x_2)] \\ \dot{x}_2 = \frac{1}{A} [q_{23}(x_1, x_2) - q_o(R_2, u_2, p_2)] \end{cases} \quad (3.37)$$

$$\begin{cases} y_1 = x_2 \\ y_2 = q_o(R_2, u_2, p_2) \\ y_3 = q_i(R_1, u_1, p_1) \end{cases} \quad (3.38)$$

onde

$$\begin{cases} q_i(R_1, u_1, p_1) = Q_{i \max} \cdot q_f(R_1, u_1, p_1) \\ q_o(R_2, u_2, p_2) = Q_{o \max} \cdot q_f(R_2, u_2, p_2) \\ q_{23}(x_1, x_2) = \frac{\sqrt{x_1 - x_2}}{R_h} \end{cases} \quad (3.39a)$$

a expressão usada para calcular a vazão q_f (Pereira, 1996) foi:

$$q_f(R_i, u_i, p_i) = \frac{1}{\sqrt{1 + \left(\frac{1}{R_i^{2(u_i-1)}} - 1 \right) \cdot p_i}} \quad (3.39b)$$

sendo $q_o(R_2, u_2, p_2)$: vazão de saída do tanque TQ-3, em m^3/s ;
 $q_i(R_1, u_1, p_1)$: vazão de entrada do tanque TQ-2, em m^3/s ;
 $q_{23}(x_1, x_2)$: vazão entre os tanques TQ-2 e TQ-3, em m^3/s ;
 $q_f(R_i, u_i, p_i)$: vazão fracionária da válvula “i”;
 Q_{imax} : vazão máxima de entrada do tanque TQ-2, em m^3/s ;
 Q_{omax} : vazão máxima de saída do tanque TQ-3, em m^3/s ;
 A : área da base dos tanques TQ-2 e TQ-3;
 R_1 : representa o alcance da válvulas FCV-1;
 R_2 : representa o alcance da válvulas FCV-2;
 p_1 : razão entre a queda de pressão mínima e máxima da válvula FCV-1;
 p_2 : razão entre a queda de pressão mínima e máxima da válvula FCV-2;
 R_h : resistência hidráulica do duto entre os tanques;
 u_1 : sinal de controle fracionário da válvula FCV-1;
 u_2 : sinal de controle fracionário da válvula FCV-2.

As variáveis de controle u_1 e u_2 indicam as posições fracionárias dos eixos das válvulas FCV-1 e FCV-2, respectivamente, ou seja indicam a abertura ou fechamento das mesmas. Estas grandezas são determinadas a partir do sistema de controle como mostrado na figura 3.24.

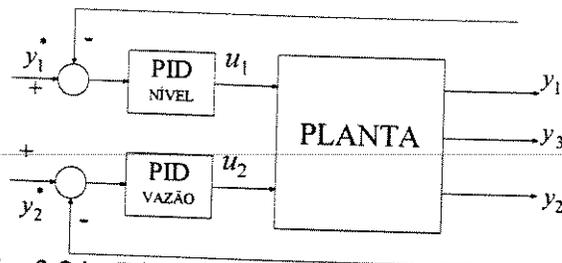


Fig. 3.24 - Diagrama de blocos do sistema de controle

3.4.3 - MODELAGEM DE FALHAS PARA O SISTEMA DE TANQUES INTERATIVOS

3.4.3.1 - Falha dos Atuadores

As falhas nos atuadores modeladas foram: *falhas nas válvulas FCV-1 e FCV-2 e nas bombas BA-1 e BA-2*. No caso das válvulas a falha mais comum é o agarramento, isto é, a válvula fica presa numa certa posição, não mais respondendo aos comandos do controlador. A modelagem desta falha é feita fixando-se o valor da vazão da válvula, que depende da posição fracionária do eixo da válvula na qual ocorreu o agarramento, u_{a1} para a válvula FCV-1 e u_{a2} para FCV-2. Portanto, tem-se:

$$q_i^{fcv1}(R_1, u_1, p_1) = Q_{imax} \cdot q_f(R_1, u_{a1}, p_1) = q_{a1} \quad (3.40)$$

para falha na válvula FCV-1 e

$$q_o^{fcv2}(R_2, u_2, p_2) = Q_{omax} \cdot q_f(R_2, u_{a2}, p_2) = q_{a2} \quad (3.41)$$

para falha em FCV-2.

Já para as bombas, a falha mais comum é a interrupção do funcionamento (por um defeito mecânico ou elétrico). Esta falha pode ser realizada desligando a bomba no painel de controle (modo manual). Para a bomba BA-1, a falha é modelada fazendo a vazão de entrada igual a zero, ou seja:

$$q_i^{fbal}(R_1, u_1, p_1) = 0 \quad (3.42)$$

para falha de desligamento da bomba BA-1.

No caso da bomba BA-2, a vazão de saída só será nula se o nível do TQ-03 for inferior à altura ao ponto de tomada de água da bomba, h_{ba2} . Caso contrário, haverá uma vazão de saída, por efeito da gravidade, até que estes níveis se igualem. Esta vazão é dependente do nível e da abertura da válvula e é dada por:

$$q_o^{fbal}(R_2, u_2, p_2, x_2) = \begin{cases} 0 & \text{se } x_2 \leq h_{ba2} \\ \frac{\sqrt{x_2 - h_{ba2}}}{R_{hb31}} \cdot R_2^{(u_2-1)} & \text{caso contrário} \end{cases} \quad (3.43)$$

Com isto, o modelo do sistema de tanques interativos que incorpora as falhas dos atuadores é dado por:

$$\begin{cases} \dot{x}_1 = \frac{1}{A} [q_i^{fa}(R_1, u_1, p_1) - q_{23}(x_1, x_2)] \\ \dot{x}_2 = \frac{1}{A} [q_{23}(x_1, x_2) - q_o^{fa}(R_2, u_2, p_2, x_2)] \end{cases} \quad (3.44)$$

$$\begin{cases} y_1 = x_2 \\ y_2 = q_o^{fa}(R_2, u_2, p_2, x_2) \\ y_3 = q_i^{fa}(R_1, u_1, p_1) \end{cases} \quad (3.45)$$

sendo

$$\begin{cases} q_i^{fa}(R_1, u_1, p_1) = \bar{k}_{fbal} \cdot Q_{imax} \cdot [\bar{k}_{fcv1} \cdot q_f(R_1, u_1, p_1) + k_{fcv1} \cdot q_{a1}] \\ q_o^{fa}(R_2, u_2, p_2, x_2) = Q_{omax} \cdot \left[\bar{k}_{fbal} \cdot (\bar{k}_{fcv2} \cdot q_f(R_2, u_2, p_2) + k_{fcv2} \cdot q_{a2}) + k_{fbal} \cdot K_{b2} \cdot \frac{\sqrt{x_2 - h_{ba2}}}{Q_{omax} \cdot R_{hb31}} \cdot R_2^{(u_2-1)} \right] \end{cases} \quad (3.46)$$

onde $\bar{k}_{fcv1} = 1 - k_{fcv1}$ e $k_{fcv1} \in \{0,1\}$ indica a falha na válvula FCV-1;

$$\begin{aligned}\bar{k}_{fcv2} &= 1 - k_{fcv2} \text{ e } k_{fcv2} \in \{0,1\} \text{ indica a falha na válvula FCV-2;} \\ \bar{k}_{fba1} &= 1 - k_{fba1} \text{ e } k_{fba1} \in \{0,1\} \text{ indica a falha na bomba BA-1;} \\ \bar{k}_{fba2} &= 1 - k_{fba2} \text{ e } k_{fba2} \in \{0,1\} \text{ indica a falha na bomba BA-2;}\end{aligned}$$

$$K_{b2} = \begin{cases} 0 & \text{se } x_2 \leq h_{ba2} \\ 1 & \text{se } x_2 > h_{ba2} \end{cases}$$

3.4.3.2 - Falha nos Componentes do Sistema

As falhas modeladas foram: *entupimentos e vazamentos* nos dutos de conexão entre tanques. Os entupimentos considerados foram: no duto de entrada de TQ-2, no duto de saída de TQ-3 e no duto de conexão TQ-2 e TQ-3. O vazamento modelado também é entre TQ-2 e TQ-3.

O *entupimento* de um duto provoca um aumento da sua resistência hidráulica e, conseqüentemente, provoca uma redução da vazão no mesmo. Os entupimentos no duto de entrada de TQ-2, no duto de saída de TQ-3 e no duto de ligação de TQ-2 e TQ-3 foram modelados multiplicando-se as vazões $q_i(R_1, u_1, p_1)$, $q_o(r_2, u_2, p_2)$ e $q_{23}(x_1, x_2)$ por parâmetros definidos no intervalo $[0, 1]$.

Já o *vazamento* entre os tanques TQ-2 e TQ-3 pode ser modelado considerando-se que o fluxo de água no vazamento do duto é composto por uma parte de origem de TQ-2 e por uma parte de origem de TQ-3. Como no sistema real esta falha é simulada abrindo a válvula FV-5, a cada uma destas componentes são atribuídas resistências hidráulicas similares à resistência entre TQ-2 e TQ-3. A vazão que escoar por razão do vazamento pode ser determinada pelo nível, h_i , de cada tanque. Com isto, tem-se o modelo do sistema para falhas nos componentes:

$$\begin{cases} \dot{x}_1 = \frac{1}{A} \left[q_i^{fc}(R_1, u_1, p_1) - q_{23}^{fc}(x_1, x_2) - \frac{\sqrt{x_1}}{R_{hv21}} \cdot k_{fv23} \right] \\ \dot{x}_2 = \frac{1}{A} \left[q_{23}^{fc}(x_1, x_2) - q_o^{fc}(R_2, u_2, p_2) - \frac{\sqrt{x_2}}{R_{hv31}} \cdot k_{fv23} \right] \end{cases} \quad (3.47)$$

$$\begin{cases} y_1 = x_2 \\ y_2 = q_o^{fc}(R_2, u_2, p_2) \\ y_3 = q_i^{fc}(R_1, u_1, p_1) \end{cases} \quad (3.48)$$

sendo

$$\begin{cases} q_i^{fc}(R_1, u_1, p_1) = \bar{k}_{fe12} \cdot Q_{imax} \cdot q_f(R_1, u_1, p_1) \\ q_o^{fc}(R_2, u_2, p_2) = \bar{k}_{fe31} \cdot Q_{imax} \cdot q_f(R_2, u_2, p_2) \\ q_{23}^{fc}(x_1, x_2) = \bar{k}_{fe23} \frac{\sqrt{x_1 - x_2}}{R_h} \end{cases} \quad (3.49)$$

onde $\bar{k}_{fe12} = 1 - k_{fe12}$ e $k_{fe12} \in [0,1]$. Indica o entupimento no duto entre TQ-1 e TQ-2;
 $\bar{k}_{fe31} = 1 - k_{fe31}$ e $k_{fe31} \in [0,1]$. Indica o entupimento no duto entre TQ-3 e TQ-1;
 $\bar{k}_{fe23} = 1 - k_{fe23}$ e $k_{fe23} \in [0,1]$. Indica o entupimento no duto entre TQ-2 e TQ-3.
 $k_{fv23} \in [0,1]$. Indica o vazamento no duto entre TQ-2 e TQ-3.

Estes parâmetros não só indicam a presença da falha mas também a sua intensidade.

3.4.3.3 - Falha de Instrumentação

As variáveis de saída y_2 e y_3 são medidas utilizando-se sensores de vazão do tipo turbina, e y_1 , sensor de nível tipo célula de carga. Para estes sensores foram consideradas somente falhas bruscas. No caso do sensor de nível, falhas bruscas podem ser causadas pela queda da linha de transmissão do sinal para o PLC ou pela desconexão do tubo de ar da cápsula da célula de carga. Já as falhas bruscas dos sensores de vazão podem ocorrer pela queda da linha de transmissão do sinal, pelo agarramento da turbina ou mesmo pela obstrução do feixe de luz infravermelho usado na medição.

Estas falhas podem ser modeladas da seguinte forma:

$$\begin{cases} \dot{x}_1 = \frac{1}{A} [q_i(R_1, u_1, p_1) - q_{23}(x_1, x_2)] \\ \dot{x}_2 = \frac{1}{A} [q_{23}(x_1, x_2) - q_o(R_2, u_2, p_2)] \end{cases} \quad (3.50)$$

$$\begin{cases} y_1 = k_{fv1} \cdot x_2 \\ y_2 = k_{fv2} \cdot q_o(R_2, u_2, p_2) \\ y_3 = k_{fv3} \cdot q_i(R_1, u_1, p_1) \end{cases} \quad (3.51)$$

onde $k_{fv1} \in \{0,1\}$ indica falha no sensor de nível;
 $k_{fv2} \in \{0,1\}$ indica falha no sensor de vazão de saída;
 $k_{fv3} \in \{0,1\}$ indica falha no sensor de vazão de entrada.

3.4.4 - MODELO GERAL DAS FALHAS NO SISTEMA DE TANQUES INTERATIVOS

Utilizando o desenvolvimento feito nos itens anteriores e incorporando algumas considerações práticas sobre a planta real, como ruído de medição e proteção (Braga, 1994; Pereira, 1996), chega-se ao modelo geral dado por (3.52) a (3.57). Com relação à proteção, esta ocorre quando o nível do tanque TQ-2 atingir o nível máximo e quando o nível do tanque TQ-3 atingir um valor de 4% do nível máximo. Para a primeira situação a bomba BA-1 é desligada. Já para a segunda situação a bomba BA-2 é desligada, provocando uma elevação do nível até 6% do máximo quando a bomba é religada.

$$\begin{cases} \dot{x}_1 = \frac{1}{A} \left[q_i^f(R_1, u_1, p_1) - q_{23}^f(x_1, x_2) - \frac{\sqrt{x_1}}{R_{hv21}} \cdot k_{fv23} \right] \\ \dot{x}_2 = \frac{1}{A} \left[q_{23}^f(x_1, x_2) - q_o^f(R_2, u_2, p_2, x_2) - \frac{\sqrt{x_2}}{R_{hv31}} \cdot k_{fv23} \right] \end{cases} \quad (3.52)$$

$$\begin{cases} y_1 = k_{fy1} \cdot x_2 + r_{y_1} \\ y_2 = k_{fy2} \cdot q_o^f(R_2, u_2, p_2, x_2) + r_{y_2} \\ y_3 = k_{fy3} \cdot q_i^f(R_1, u_1, p_1) + r_{y_3} \end{cases} \quad (3.53)$$

sendo

$$\begin{cases} q_i^f(R_1, u_1, p_1) = Q_{imax} \bar{k}_{fe12} \cdot \bar{k}_{fba1} \left[\bar{k}_{fcv1} \cdot q_f(R_1, u_1, p_1) + k_{fcv1} \cdot q_{a1} \right] \\ q_o^f(R_2, u_2, p_2, x_2) = Q_{omax} \bar{k}_{fe31} \cdot \left[\bar{k}_{fba2} \cdot \left(\bar{k}_{fcv2} \cdot q_f(R_2, u_2, p_2) + k_{fcv2} \cdot q_{a2} \right) - k_{fba2} \cdot K_{b2} \cdot \frac{\sqrt{x_2 - h_{hba2}}}{Q_{omax} \cdot R_{hb31}} \cdot R_2^{(u_2 - 1)} \right] \\ q_{23}^f(x_1, x_2) = \bar{k}_{fe23} \cdot \frac{\sqrt{x_1 - x_2}}{R_h} \\ q_f(R_i, u_i, p_i) = \frac{1}{\sqrt{1 + \left(\frac{1}{R_i^{2(u_i - 1)}} - 1 \right) \cdot p_i}} \end{cases} \quad (3.54)$$

onde r_{y_1} , r_{y_2} e r_{y_3} são variáveis aleatórias de distribuição uniforme, média nula e amplitude máxima $a_1 = 0,075\% \cdot x_{2max}$, $a_2 = 0,25\% Q_{omax}$ e $a_3 = 0,5\% Q_{imax}$, respectivamente. Estas variáveis foram incluídas no modelo para simular o efeito do ruído de medição observado (Pereira, 1996).

As condições de intertravamento são modeladas pelas constantes da seguinte forma:

$$k_{fba1} = \begin{cases} 1 & \text{se } x_1 > x_{1max} \text{ ou se ocorrer falha na bomba BA1} \\ 0 & \text{se } x_1 \leq x_{1max} \text{ se não ocorrer falha na bomba BA1} \end{cases} \quad (3.55)$$

$$k_{fba2} = \begin{cases} 1 & \text{se } x_2 \leq 0.04 \cdot x_{2max} \text{ ou se ocorrer falha na bomba BA2} \\ 0 & \text{se } x_2 \geq x_{2max} \text{ se não ocorrer falha na bomba BA2} \end{cases} \quad (3.56)$$

$$K_{b2} = \begin{cases} 0 & \text{se } x_2 \leq h_{ba2} \\ 1 & \text{se } x_2 > h_{ba2} \end{cases} \quad (3.57)$$

Com este modelo é possível simular as diversas falhas resumidas na tabela 3.

Tabela 3: Tipo de falhas no sistema de tanques interativos

Tipo de Falha	Parâmetro Indicador da Falha	definição do Parâmetro
agarramento da válvula CV-1	k_{fcv1}	{0,1}
agarramento da válvula CV-2	k_{fcv2}	{0,1}
desligamento da bomba BA-01	k_{ba1}	{0,1}
desligamento da bomba BA-02	k_{ba2}	{0,1}
entupimento no duto entre TQ-1 e TQ-2	k_{fe12}	[0,1]
entupimento no duto entre TQ-3 e TQ-1	k_{fe31}	[0,1]
entupimento no duto entre TQ-2 e TQ-3	k_{fe23}	[0,1]
vazamento no duto entre TQ-2 e TQ-3	k_{fv23}	[0,1]
falha no sensor de nível de TQ-3	k_{fy1}	{0,1}
falha no sensor de vazão de saída de TQ-3	k_{fy2}	{0,1}
falha no sensor de vazão de entrada de TQ-2	k_{fy3}	{0,1}

3.4.5 - RESULTADOS DE SIMULAÇÕES DE FALHAS NO SISTEMA DE TANQUES INTERATIVOS

A figura 3.25 mostra as curvas das variáveis de entrada, sinais de controle, e saída do sistema de tanques interativos quando ocorre um entupimento no duto de conexão de TQ-3 para TQ-1 em $t=500$ segundos. A intensidade do entupimento é $k_{fe31}=0,79$. Na prática essa situação pode ser efetuada através do fechamento da válvula FV-7. Quando ocorre o entupimento o controlador de vazão de saída de TQ-3 tende a compensar o efeito abrindo mais a válvula FCV-2. Para este caso simulado, a vazão máxima de água que pode fluir na tubulação é inferior à vazão desejada para compensar o efeito do entupimento, valor do *setpoint*, com isto a malha de vazão não é controlada. Nota-se que a vazão de entrada de TQ-2 também sofreu redução, para permitir o controle de nível de TQ-3, via fechamento da válvula FCV-1. Para entupimentos de menor intensidade, por exemplo $k_{fe31}=0,11$, a vazão de saída de TQ-3 não é alterada, figura 3.26. A figura 3.27 mostra a situação onde ocorre uma falha no sensor de

vazão de saída, interrupção total do sinal transmitido do sensor para o PLC. Após a ocorrência da falha a leitura do sensor de vazão FE-2 é nula. Isto faz com que o controlador de vazão abra a válvula FCV-1 para tentar fazer com que a vazão medida seja igual ao *setpoint*. Isto acarreta um aumento da vazão de saída real, que não é o valor medido, e conseqüentemente aumenta a vazão de entrada de TQ-2, para efetuar o controle do nível de TQ-3. Outros resultados simulados com o modelo são apresentados em (Pereira, 1996; Camargo, 1997), onde também são apresentados resultados experimentais obtidos na planta piloto.

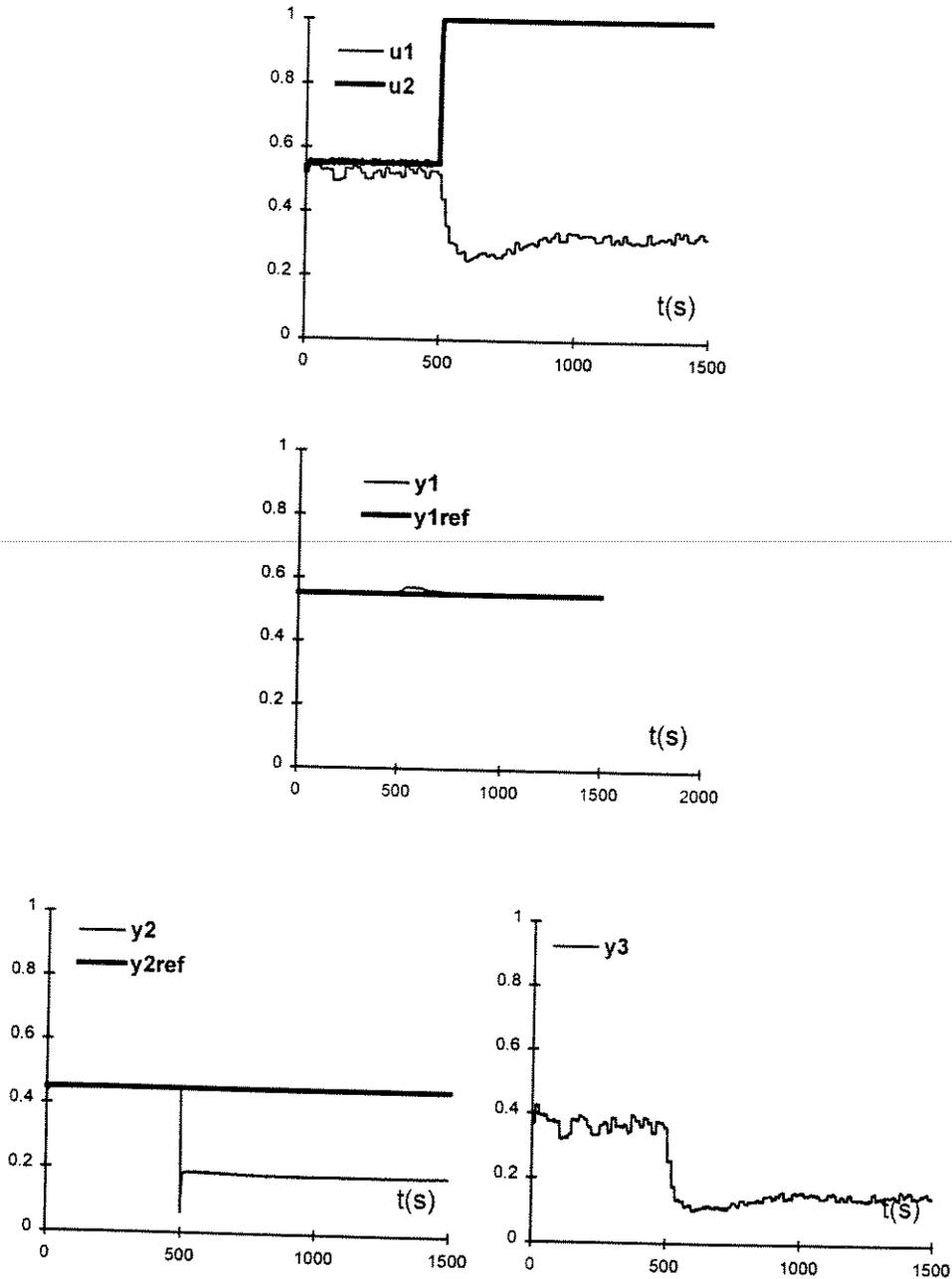


Fig. 3.25 - Curvas das variáveis de entrada e saída do STI para entupimento entre TQ-3 e TQ-1 com $k_{fe31}=0.79$.

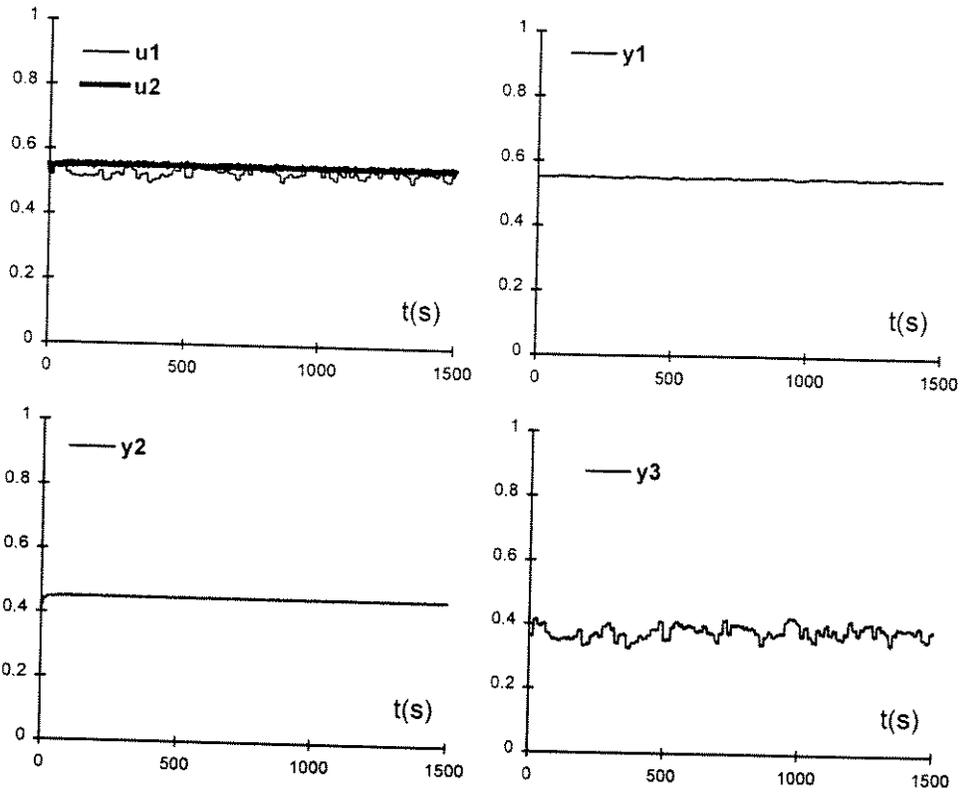


Fig. 3.26 - Curvas das variáveis de entrada e saída do STI para entupimento entre TQ-3 e TQ-1 com $k_{fe31}=0.11$.

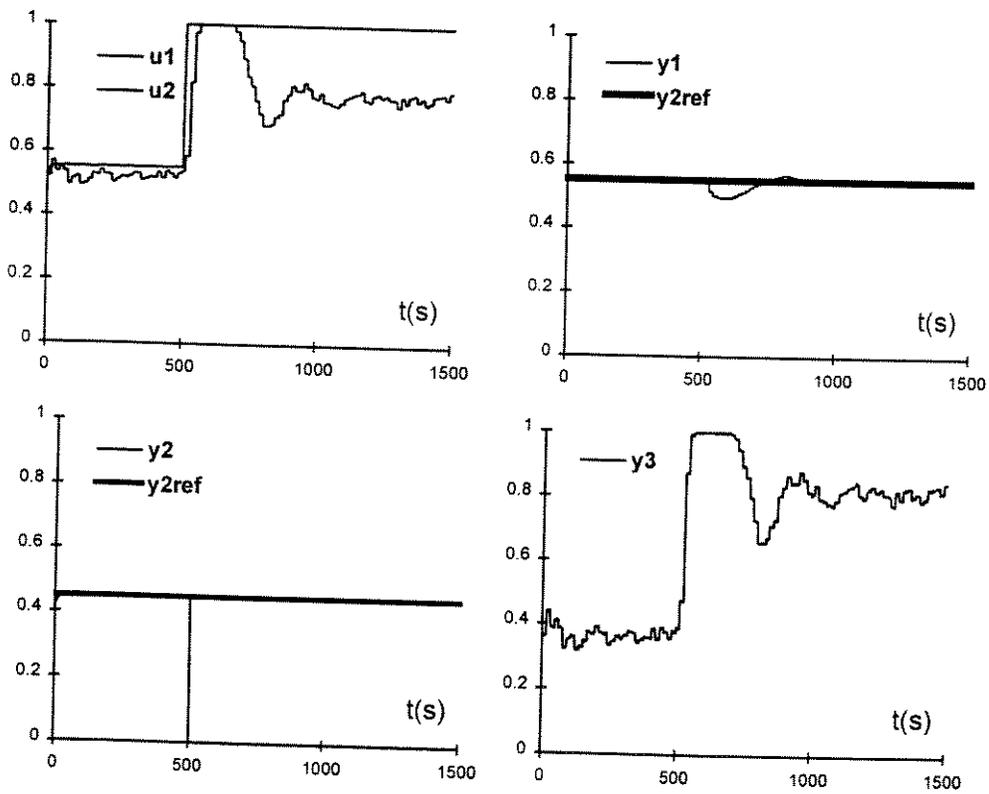


Fig. 3.27 - Curvas das variáveis de saída e entrada do STI para falha no sensor de vazão de saída de TQ-3 e TQ-1.

3.5 - CONCLUSÃO

Neste capítulo foram desenvolvidos modelos matemáticos que permitem simular falhas em três sistemas dinâmicos não lineares. Estes modelos serão utilizados para projetar e testar os sistemas de detecção e diagnóstico de falhas propostos neste trabalho. A vantagem em se ter um modelo bem próximo da planta real é que o projeto dos algoritmos podem ser feito via simulação, necessitando apenas de pequenos ajustes na implementação real. Para o sistema de tanques interativos os resultados de simulações foram comparados com resultados experimentais e o erro é bastante pequeno (Pereira, 1996). Mais resultados de simulações dos modelos serão apresentados nos capítulo seguintes.

CAPÍTULO 4

OBSERVADORES DE ESTADOS PARA SISTEMAS INCERTOS: ABORDAGEM BASEADA NA TEORIA DE MODOS DESLIZANTES

4.1 - INTRODUÇÃO

O tema tratado neste capítulo é observadores de estados em sistemas dinâmicos incertos. Primeiramente é apresentada a estrutura de um observador para sistemas discretos incertos, proposta por (Takahashi e Peres, 1997). Tal observador é fundamentado na teoria de modos deslizantes (*sliding modes*) (Utkin, 1984). O observador Discreto em Modos Deslizantes (ODMD) é robusto quanto a perturbações paramétricas e entradas e/ou parâmetros desconhecidos. Este observador é então empregado na estimação dos estados do sistema de acionamento C.C., modelado no capítulo anterior. São projetados observadores de estado para o sistema de acionamento C.C., que rejeitam perturbações relacionadas às direções dos estados do modelo. O desenvolvimento deste capítulo será utilizado na estratégia de detecção e diagnóstico de falhas, apresentadas no capítulo 1. A vantagem do observador em modos deslizantes (OMD) é a simplicidade de projeto e bom desempenho (Caminhas *et alii*, 1996^e), como será mostrado nos resultados de simulações.

4.2 - OBSERVADOR EM MODOS DESLIZANTES P/ SISTEMAS DISCRETOS NÃO LINEARES INCERTOS

Conforme Takahashi e Peres (1997), parte-se do seguinte sistema discreto não-linear:

$$\begin{aligned}x(k+1) &= F(x(k), u(k), k) + \Delta F(x(k), u(k), k) + D.v(k) \\ y(k) &= C.x(k)\end{aligned}\tag{4.1}$$

onde $x \in \mathbf{R}^n$, $u \in \mathbf{R}^p$, $y \in \mathbf{R}^m$, $v \in \mathbf{R}^q$, $k \in \mathbf{N}$. Neste sistema, x representa o vetor de estados, que irá ser estimado pelo observador, u representa o vetor de entradas de controle (suposto conhecido), v representa um vetor de entradas desconhecidas (distúrbios), e y representa um vetor de medições disponível. A dinâmica do sistema é composta por uma parte “nominal” F e por uma parte “perturbada” ΔF . Supõem-se conhecidas a função $F(\cdot, \cdot, \cdot)$ e a matriz C , e desconhecidas a matriz D e a função $\Delta F(\cdot, \cdot, \cdot)$, obedecendo porém a seguinte condição de “matching” das perturbações:

$$\begin{aligned}\mathfrak{R}(H) &= \mathfrak{R}(D) \cup F \\ F &= \text{Im}(\Delta F)\end{aligned}\tag{4.2}$$

sendo $\mathfrak{R}(\cdot)$ o espaço imagem da matriz argumento, e $\text{Im}(\cdot)$ o espaço imagem da função argumento. A matriz H em (4.2) é suposta conhecida e indica as direções onde existem

perturbações ou entradas desconhecidas. Além disso, a matriz H deve satisfazer conjuntamente com C :

$$\begin{aligned}\rho(C.H) &= \rho(H) = r \\ \rho(C) &= m \geq r\end{aligned}\tag{4.3}$$

onde $\rho(\cdot)$ representa o rank da matriz argumento e r é o número de direções onde aparecem perturbações que serão desacopladas. Como última suposição, a planta nominal (sistema 4.1, excluídas as entradas desconhecidas) deve ser estável e observável através da matriz C em toda a faixa de operação.

Com a condição (4.2), pode-se escrever:

$$\begin{aligned}\forall(x, k, u, v) \exists \omega : \\ H.\omega &= \Delta F(x(k), u(k), k) + D.v(k)\end{aligned}\tag{4.4}$$

Isto permite rescrever a equação (4.1) como:

$$x(k+1) = F(x(k), u(k), k) + H.\omega(k)\tag{4.5}$$

O observador discreto em modos deslizantes é construído acrescentando-se a (4.5) um termo de cancelamento de perturbações $d(k)$:

$$\hat{x}(k+1) = F(\hat{x}(k), u(k), k) + H.\omega(k) + H.d(k)\tag{4.6}$$

Considera-se a seguinte partição do vetor de saída:

$$y(k) = C.x(k) = \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = \begin{bmatrix} C^1 \\ C^2 \end{bmatrix}.x(k)\tag{4.7}$$

onde a partição y^1 do vetor de saída contém as medidas usadas para viabilizar o desacoplamento das perturbações.

O termo $d(k)$ é deduzido a partir da restrição:

$$C^1.\hat{x}(k+1) - y^1(k+1) = 0\tag{4.8}$$

A restrição (4.8) equivale a uma redução de ordem do sistema, ou a impor que a evolução temporal do vetor de estados pertença a uma superfície do espaço de estados $C^1.e(k)=0$, onde $e(k)=\hat{x}(k)-x(k)$. Essa superfície é chamada “superfície de deslizamento”, e o observador é dito estar em “modos deslizantes” nessa superfície, por analogia com os sistemas de controle em modos deslizantes (Utkin, 1984; De Carlo *et alii*, 1988).

Substituindo (4.8) em (4.6) conduz a:

$$\hat{x}(k+1) = [I - H.(C^1 H)^{-L} C^1] . F(\hat{x}(k), u(k), k) + H.(C^1 H)^{-L} y^1(k+1) \quad (4.9)$$

onde $(\cdot)^{-L}$ significa qualquer inversa à esquerda da matriz argumento.

Finalmente, para aumentar a liberdade na seleção da dinâmica do erro, acrescenta-se à equação (4.9) um termo proporcional ao erro de saída, à semelhança do clássico observador de Luenberger (1971):

$$\hat{x}(k+1) = [I - H.(C^1 H)^{-L} C^1] . F(\hat{x}(k), u(k), k) + H.(C^1 H)^{-L} y^1(k+1) + L.[C^r . \hat{x}(k) - y^r(k)] \quad (4.10)$$

sendo y^r uma partição do vetor u que contém as saídas que serão realimentadas e C^r a correspondente partição da matriz C .

Omitindo os índices dos vetores correspondentes ao instante k e considerando:

$$\begin{aligned} \Phi_2 &= H.(C^1 H)^{-L} \\ \Phi_1 &= I - \Phi_2 . C^1 \end{aligned} \quad (4.11)$$

tem-se a seguinte expressão para o observador:

$$\hat{x}(k+1) = \Phi_1 . F(\hat{x}, u, k) + \Phi_2 . y^1(k+1) + L.[C^r . \hat{x}(k) - y^r(k)] \quad (4.12)$$

Esta equação corresponde a um observador capaz de rejeitar completamente perturbações conforme descritas em (4.4), com uma dinâmica de erro selecionável. A análise geral da convergência do erro não será apresentada aqui. No entanto, é fácil ver que, para o caso particular (suficiente para os propósitos deste trabalho) em que F é globalmente assintoticamente estável em relação a um ponto de equilíbrio e u é constante (ou constante por partes), pode-se sempre escolher os parâmetros de projeto de maneira a garantir a convergência global do observador. O projetista deve selecionar as matrizes H , L e $(C^1 H)^{-L}$, procurando um compromisso adequado entre a dinâmica do erro e a rejeição de perturbações, como discutido em (Takahashi e Peres, 1997).

Para sistemas bilineares, como é o caso do sistema de acionamento C.C., tem-se que a função $F(x, u, k)$ pode ser escrita como:

$$F(x, u, k) = A(x).x + B.u \quad (4.13)$$

Escrevendo A no lugar de $A(x)$, para simplificar a notação, tem-se a seguinte equação do observador:

$$\hat{x}(k+1) = \Phi_F . \hat{x}(k) + \Phi_1 . B . u(k) + \Phi_2 . y^1(k+1) - L . y^r(k) \quad (4.14)$$

sendo $\Phi_F = [\Phi_1 . A + L . C^r]$ a matriz de transição em malha fechada.

O projeto deste observador, que é desenvolvido algebricamente, é simples, bastando seguir os passos:

1. construa a matriz H ;
2. escolha a partição da matriz C ;
3. calcule $C^l.H$;
4. encontre uma inversa à esquerda para $C^l.H$;
5. calcule Φ_2 e Φ_1 , expressão (4.11);
6. escolha a matriz de ganhos do observador;
7. estabeleça a equação final do observador.

4.3 - PROJETO DOS OBSERVADORES EM MODOS DESLIZANTES PARA O SISTEMA DE ACIONAMENTO C.C.

Neste item serão mostrados os passos detalhados do projeto do conjunto de observadores para o sistema de acionamento C.C., modelado no capítulo 2. O modelo do sistema, não considerando falhas, é expresso por (4.15). Pode-se demonstrar que o sistema é estável e observável; portanto é possível aplicar a teoria apresentada no item anterior.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} a_1 & a_2(k) & 0 \\ 0 & a_3 & 0 \\ a_4(k) & 0 & a_5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} [T_L(k)] \quad (4.15)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

Verifica-se pela equação (4.3), que o número de direções em que as perturbações podem ser desacopladas é menor ou igual ao número de medições efetuadas. Além disso, para ser possível uma alocação arbitrária dos autovalores do observador, este número deve ser estritamente menor que o número de medições. Por isto, para o sistema (4.15), serão projetados observadores que rejeitam perturbações em uma ou duas direções do espaço das variáveis de estado. Serão projetados seis observadores abreviados por O^1 , O^2 , O^3 , O^{12} , O^{13} e O^{23} . O sobrescrito indexa as direções das perturbações que serão rejeitadas.

4.3.1 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NA DIREÇÃO DE x_1

Para este primeiro projeto serão seguidos todos os passos do algoritmo mostrado anteriormente.

- **Passo 1**

No caso do sistema (4.15), a matriz H é construída da seguinte forma:

$$H = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (4.16)$$

A matriz H é formada por “0” e “1”, sendo que a presença do elemento de valor “1” indica a direção onde aparecem perturbações desconhecidas (variações paramétricas ou entradas desconhecidas), a qual deseja-se desacoplar. Como pode ser verificado, com esta matriz H , juntamente com a matriz C , a condição (4.3) é satisfeita.

- **Passo 2**

A partição da matriz C adotada para este observador foi:

$$\begin{bmatrix} y^1(k) \\ \dots \\ y^2(k) \end{bmatrix} = \begin{bmatrix} y_1(k) \\ \dots \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} C^1 \\ \dots \\ C^2 \end{bmatrix} \cdot x(k) = \begin{bmatrix} 1 & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \quad (4.17)$$

portanto $C^1 = [1 \ 0 \ 0]$. Qualquer outra opção de partição de C implicaria numa matriz $C^1 \cdot H$ nula, inviabilizando o projeto do observador.

- **Passo 3**

$$C^1 \cdot H = [1 \ 0 \ 0] \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = [1] \quad (4.18)$$

- **Passo 4**

Neste caso a inversa à esquerda é a própria inversa, já que $C^1 \cdot H$ é uma matriz quadrada, ou seja

$$(C^1 \cdot H)^{-L} = [1] \quad (4.19)$$

- *Passo 5*

$$\phi_2 = H.(C'.H)^{-L} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot [1] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (4.20)$$

$$\phi_1 = I - \phi_2 \cdot C' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot [1 \ 0 \ 0] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

- *Passo 6*

A dinâmica de erro do observador depende dos autovalores da matriz $\Phi_F = \Phi_1 \cdot A + L \cdot C'$, que podem ser arbitrariamente alocados pela escolha de L . Tomando o caso mais geral, que é todas as saídas sendo realimentadas, tem-se:

$$L = \begin{bmatrix} L_{11}(k) & L_{12}(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k) & L_{23}(k) \\ L_{31}(k) & L_{32}(k) & L_{33}(k) \end{bmatrix} \text{ e } C' = C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

logo

$$\Phi_F = \begin{bmatrix} L_{11}(k) & L_{12}(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k) + a_3 & L_{23}(k) \\ L_{31}(k) + a_4(k) & L_{32}(k) & L_{33}(k) + a_5 \end{bmatrix} \quad (4.23)$$

Para casos particulares, onde uma ou duas saídas são realimentadas, basta desconsiderar as colunas de L que correspondem às saídas não realimentadas.

Os valores dos ganhos de L podem ser determinados a partir da expressão:

$$|\lambda \cdot I - \Phi_F| = 0 \quad (4.24)$$

Considerando que o sistema possui três autovalores, a partir de (4.24) é possível montar três equações envolvendo os nove ganhos. Para resolver esse sistema de equações é necessário arbitrar no mínimo seis ganhos para calcular os demais. Por exemplo, arbitrando os ganhos $L_{ij}(k) = 0$ para $i \neq j$ é possível determinar os demais facilmente a partir dos autovalores desejados:

$$[\lambda \cdot I - \Phi_F] = \begin{bmatrix} \lambda - L_{11}(k) & 0 & 0 \\ 0 & \lambda - L_{22}(k) - a_3 & 0 \\ -a_4(k) & 0 & \lambda - L_{33}(k) - a_5 \end{bmatrix} \quad (4.25)$$

Portanto, a alocação arbitrária de autovalores é facilmente realizada fazendo-se:

$$L_{11}(k) = L_{11} = \lambda_1; \quad L_{22}(k) = L_{22} = \lambda_2 - a_3 \quad \text{e} \quad L_{33}(k) = L_{33} = \lambda_3 - a_5 \quad (4.26)$$

sendo λ_1 , λ_2 e λ_3 os autovalores desejados.

• **Passo 7**

Para estabelecer a equação resta calcular os termos:

$$\Phi_1 \cdot B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \quad (4.27)$$

$$\Phi_2 \cdot y^1(k+1) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [y_1(k+1)] \quad (4.28)$$

Finalmente, a equação do observador é:

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} + a_3 & 0 \\ a_4(k) & 0 & L_{33} + a_5 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.29)$$

$$+ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [y_1(k+1)] - \begin{bmatrix} L_{11} \cdot y_1(k) \\ L_{22} \cdot y_2(k) \\ L_{33} \cdot y_3(k) \end{bmatrix}$$

Observa-se que as equações são bastante simples, envolvendo apenas operações básicas, tornando o observador atraente em termos de implementação prática.

4.3.2 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NA DIREÇÃO DE x_2

Para rejeição de perturbações na direção de x_2 a matriz H é:

$$H = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (4.30)$$

A partição da matriz C para este observador é $C^1 = [0 \ 1 \ 0]$. Com isto, $y^1(\cdot) = [y_2(\cdot)]$.

Seguindo todos os passos do algoritmo, chega-se à expressão final do observador para rejeição de perturbações na direção de x_2 :

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k)+a_1 & L_{12}(k)+a_2(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k) & L_{23}(k) \\ L_{31}(k)+a_4(k) & L_{32}(k) & L_{33}(k)+a_5 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.31)$$

$$+ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cdot [y_2(k+1)] - [L][y'(k)]$$

4.3.3 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NA DIREÇÃO DE x_3

Para rejeição de perturbações na direção de x_3 a matriz H é:

$$H = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.32)$$

A partição da matriz C para este observador é $C^1 = [0 \ 0 \ 1]$.

Seguindo todos os passos do algoritmo, chega-se ao à expressão final do observador para rejeição de perturbações na direção de x_3 :

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k)+a_1 & L_{12}(k)+a_2(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k)+a_3 & L_{23}(k) \\ L_{31}(k) & L_{32}(k) & L_{33}(k) \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.33)$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot [y_3(k+1)] - [L][y'(k)]$$

4.3.4 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NAS DIREÇÕES DE x_1 E x_2

Para rejeição de perturbações nas direções de x_1 e x_2 tem-se:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (4.34)$$

$$C^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.35)$$

$$y^i(\cdot) = [y_1(\cdot) \quad y_2(\cdot)]^T \quad (4.36)$$

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k) & L_{12}(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k) & L_{23}(k) \\ L_{31}(k)+a_4(k) & L_{32}(k) & L_{33}(k)+a_5 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.37)$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} - [L][y^r(k)]$$

4.3.5 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NAS DIREÇÕES DE x_1 E x_3

Para rejeição de perturbações nas direções de x_1 e x_3 tem-se:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.38)$$

$$C^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.39)$$

$$y^i(\cdot) = [y_1(\cdot) \quad y_3(\cdot)]^T \quad (4.40)$$

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k) & L_{12}(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k)+a_3 & L_{23}(k) \\ L_{31}(k) & L_{32}(k) & L_{33}(k) \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.41)$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(k+1) \\ y_3(k+1) \end{bmatrix} - [L][y^r(k)]$$

4.3.6 - OBSERVADOR PARA REJEIÇÃO DE PERTURBAÇÕES NAS DIREÇÕES DE x_2 E x_3

Para rejeição de perturbações nas direções de x_2 e x_3 tem-se:

$$H = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.42)$$

$$C^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.43)$$

$$y^1(\cdot) = [y_2(\cdot) \quad y_3(\cdot)]^T \quad (4.44)$$

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k)+a_1 & L_{12}(k)+a_2(k) & L_{13}(k) \\ L_{21}(k) & L_{22}(k) & L_{23}(k) \\ L_{31}(k) & L_{32}(k) & L_{33}(k) \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} \quad (4.45)$$

$$+ \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_2(k+1) \\ y_3(k+1) \end{bmatrix} - [L][y^1(k)]$$

4.4 - RESULTADOS DE SIMULAÇÕES

Para avaliar o desempenho do observador em modos deslizantes, foram utilizados os modelos do sistema considerando falhas, expressão (3.12), e do observador O^1 , expressão (4.29), realimentando somente a saída y_1 . Neste caso, somente um autovalor pode ser realocado. Os autovalores do observador, para as simulações apresentadas, são $[a_1/2 \quad a_3 \quad a_3]$. As situações apresentadas correspondem a: curto-circuito do conversor da armadura; curto-circuito do conversor do campo e variação de carga (50 % da nominal de sobrecarga), correspondendo a perturbações nas direções de x_1 , x_2 e x_3 , respectivamente. Como era esperado, o observador estima de forma correta quando há uma perturbação na direção de x_1 , figuras 4.1, 4.2 e 4.3, e apresenta erro de estimação para as outras duas situações, figuras de 4.4 a 4.9. Para todos os gráficos, as grandezas estão normalizadas, tendo como bases os valores nominais.

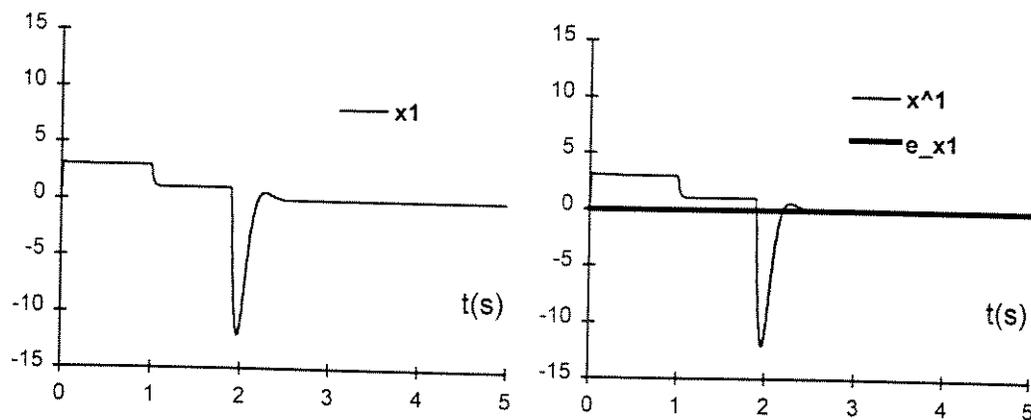


Fig. 4.1 - Erro do estado x_1 do observador O^1 quando ocorre erro na direção de x_1

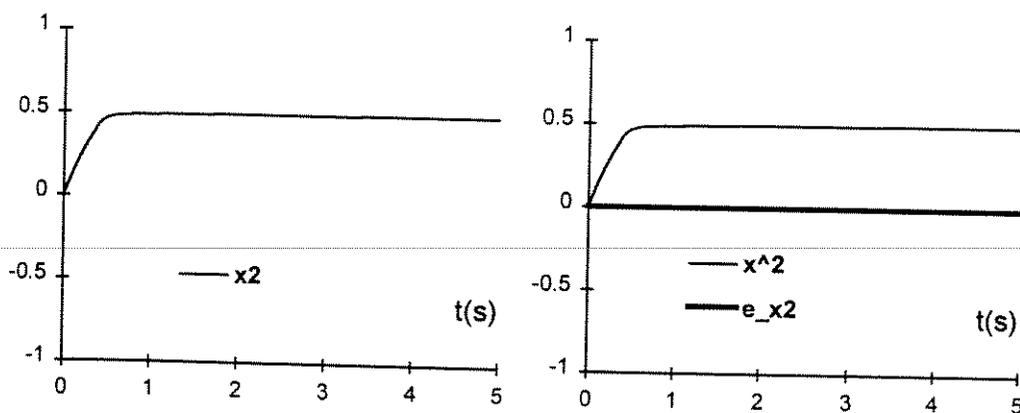


Fig. 4.2 - Erro do estado x_2 do observador O^1 quando ocorre erro na direção de x_1

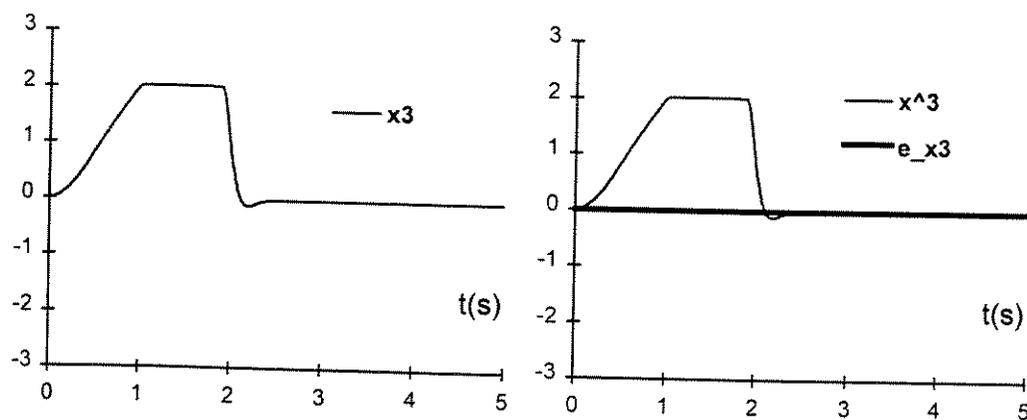


Fig. 4.3 - Erro do estado x_3 do observador O^1 quando ocorre erro na direção de x_1

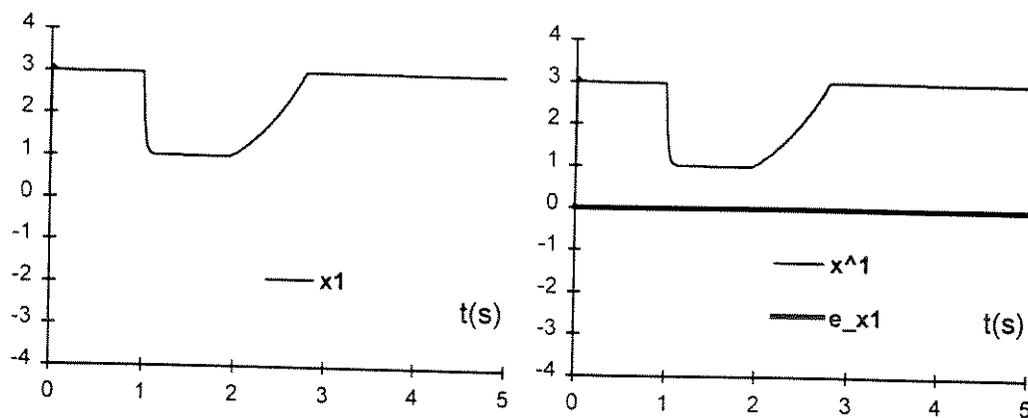


Fig. 4.4 - Erro do estado x_1 do observador O^1 quando ocorre erro na direção de x_2

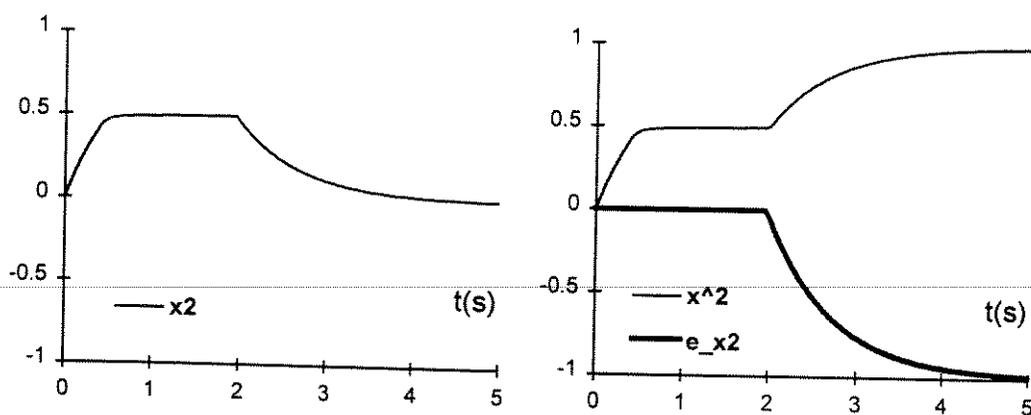


Fig. 4.5 - Erro do estado x_2 do observador O^1 quando ocorre erro na direção de x_2

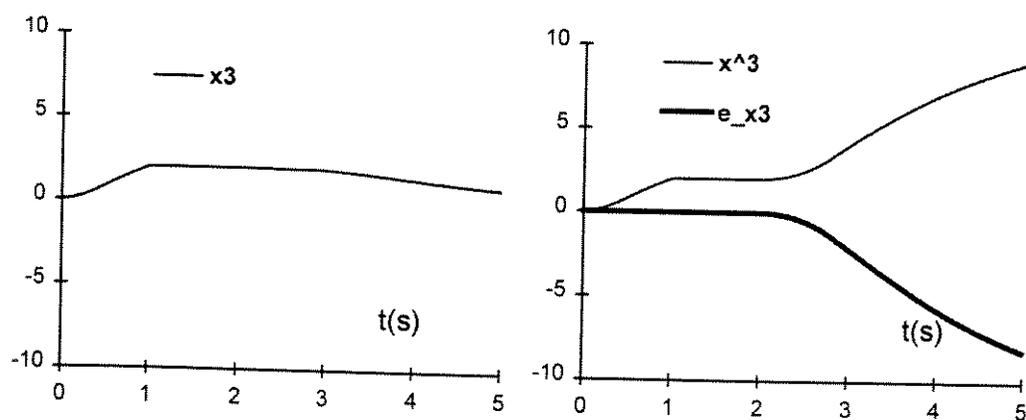


Fig. 4.6 - Erro do estado x_3 do observador O^1 quando ocorre erro na direção de x_2

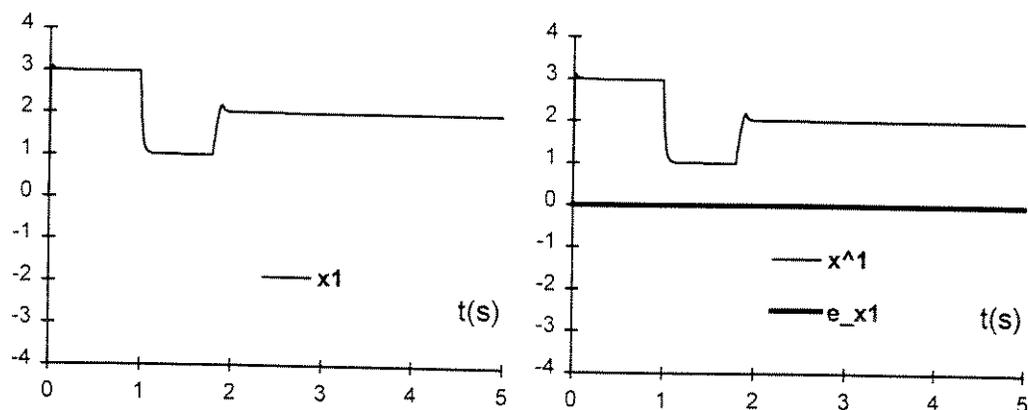


Fig. 4.7 - Erro do estado x_1 do observador O^1 quando ocorre erro na direção de x_3

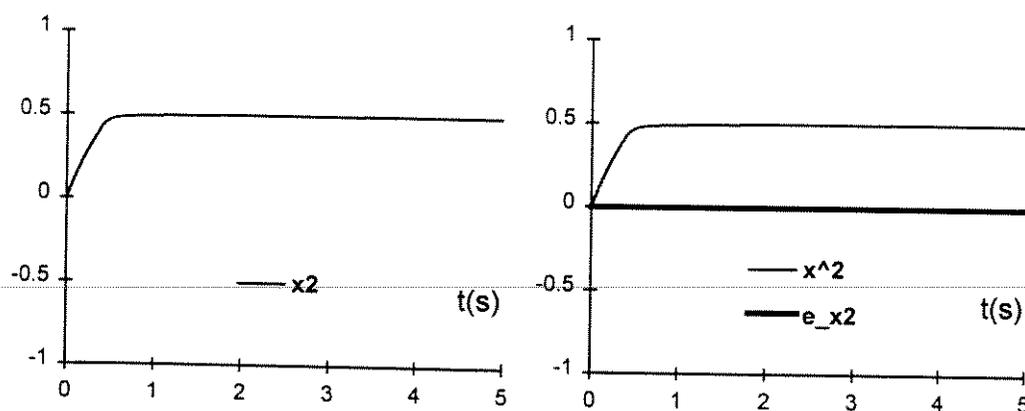


Fig. 4.8 - Erro do estado x_2 do observador O^1 quando ocorre erro na direção de x_3

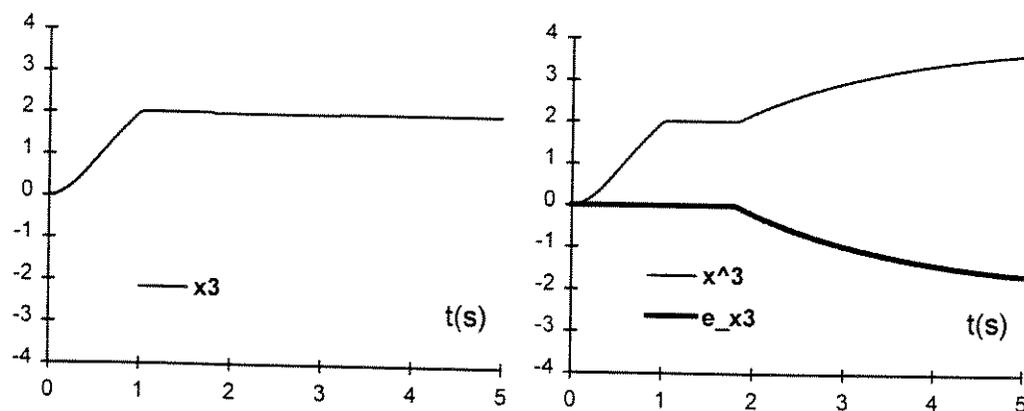


Fig. 4.9 - Erro do estado x_3 do observador O^1 quando ocorre erro na direção de x_3

4.5 - CONCLUSÃO

Este capítulo apresentou uma estrutura de observador de estados para sistemas discretos incertos (Takahashi e Peres, 1995). Tal abordagem foi aplicada na estimação de estados do sistema de acionamento C.C.. Foram desenvolvidos três observadores para rejeição de perturbações em uma ou duas direções dos três estados do sistema. Os modelos obtidos são bastante convenientes, pois envolvem um pequeno número de operações de soma e multiplicação. Além disso, a alocação dinâmica dos autovalores do observador é feita de forma bastante simples.

As vantagens da estrutura utilizada são:

- a simplicidade de projeto;
- possibilidade de se trabalhar com parâmetros desconhecidos, que é mais forte que uma simples rejeição de variação paramétrica ou rejeição de incertezas.

CAPÍTULO 5

IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS USANDO

NEO FUZZY NEURON

5.1 - INTRODUÇÃO

O problema de identificação de sistemas dinâmicos pode ser encarado como mapeamento de um espaço de entrada para um espaço de saída. Devido à capacidade das redes neurais artificiais, de múltiplas camadas, de aproximação universal de mapeamentos não-lineares arbitrários (Hornik *et alii*, 1990; Chang e Abdel-Ghaffar, 1992; Chen e Chen, 1995) estas redes vêm sendo utilizadas em identificação de sistemas dinâmicos lineares ou não-lineares (Narendra e Parthasarathy, 1990; Zuben, 1993; Caminhas *et alii*, 1994; Zhang e Fadali, 1996). Das topologias de redes neurais artificiais a rede tipo perceptron multicamadas é a mais utilizada em identificação de sistemas não-lineares.

Uma outra categoria de rede, que também tem capacidade de mapeamento não-linear, engloba as redes *neurofuzzy*. Essas redes utilizam características *fuzzy* para a modificação das conexões sinápticas. Neste sentido, em Wang (1994) é proposto um sistema *fuzzy* adaptativo, onde as regras são implementadas por uma rede *neurofuzzy*. É empregado o algoritmo de *backpropagation* para ajustar os parâmetros das funções de pertinência, associadas aos antecedentes e consequentes das regras *fuzzy*.

Em Yamakawa *et alii* (1992) é proposto um neurônio *fuzzy*, batizado por “neo *fuzzy neuron*” (NFN), que do ponto de vista computacional apresenta tempo de processamento bastante inferior às estruturas de rede neural multicamadas e rede *neurofuzzy* proposta por Wang (1994).

Neste capítulo é empregada a estrutura proposta por Yamakawa *et alii* (1992), com algumas modificações, em identificação de sistemas dinâmicos não-lineares. Tais modificações visam generalizar a estrutura original com relação ao número de saídas e número de partições *fuzzy* diferente para cada entrada. A escolha dessa topologia está relacionada à rapidez de processamento, fator importante para futura implementação em tempo real, aliada ao bom desempenho. São apresentados 4 exemplos, sendo que 3 deles são os mesmos utilizados em Narendra e Parthasarathy (1990) e Wang (1994). Isto permite fazer uma análise comparativa entre as 3 estruturas.

5.2 - NEO FUZZY NEURON

5.2.1 - MOTIVAÇÃO

A capacidade de mapeamentos complexos das redes neurais cresce com o número de camadas e neurônios. Entretanto, este aumento acarreta maior tempo de processamento, que é um fator importante em implementação em tempo real. Além disso, o treinamento das redes neurais multicamadas, utilizando algoritmo *backpropagation*, consiste na minimização de uma função não-linear. A convergência dos métodos de otimização não-linear depende da

condição inicial e do comportamento da função objetivo (Mateus e Luna, 1986). Para o treinamento das redes neurais estes dois pontos são críticos, pois a condição inicial é atribuída aleatoriamente e a função objetivo não é estritamente convexa, apresentando mínimos locais (Yamakawa *et alii*, 1992). Neste caso, a convergência não é assegurada, podendo às vezes apresentar um erro significativo mesmo após longo tempo de treinamento (Nunes e Gomide, 1996).

Uma proposta que surgiu com intuito de solucionar o problema do mínimo local é o neurônio *fuzzy* de Yamakawa *et alii* (1992), batizado por *neo fuzzy neuron* (NFN). Isto porque a função objetivo a ser minimizada no processo de treinamento é quadrática e convexa, como mostrado posteriormente.

5.2.2 - ESTRUTURA DO NFN

A estrutura do NFN é mostrada na figura 5.1. As características de cada sinápsse são representadas por uma função não-linear $f_i(x_i)$, determinada a partir de inferência *fuzzy* e *defuzzificação*. A agregação dos sinais sinápticos deixa de ser caracterizada por uma função sigmoideal, como nas redes convencionais, passando a ser obtida simplesmente por uma soma algébrica (Yamakawa *et alii*, 1992; Nunes e Gomide, 1996). Isto faz com que a função erro do NFN seja quadrática e convexa, com relação aos pesos que são ajustados durante o seu treinamento; portanto, o mínimo local é mínimo global (Yamakawa *et alii*, 1992).

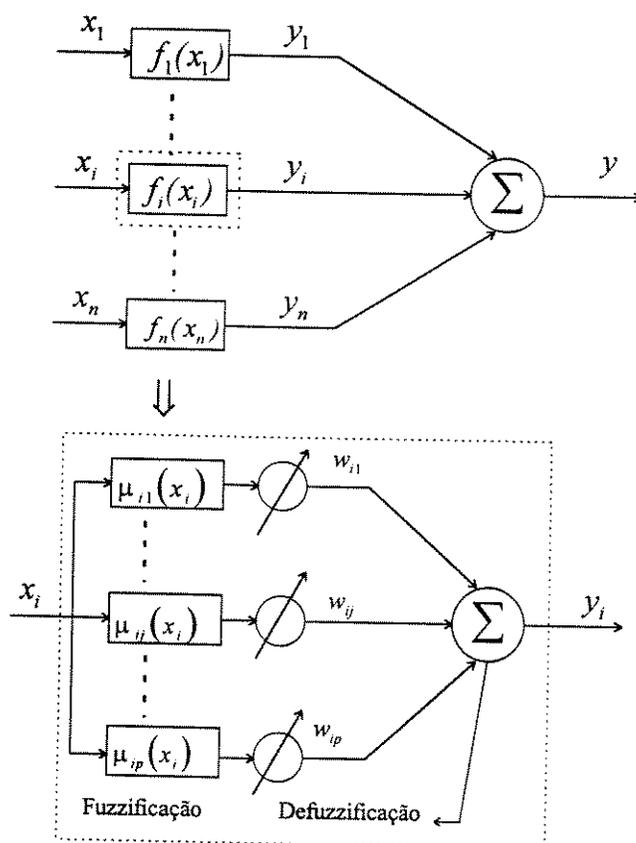


Fig. 5.1 - Estrutura do NFN

A saída do NFN pode ser representada pela seguinte equação:

$$y = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) = \sum_{i=1}^n y_i(x_i) = \sum_{i=1}^n f_i(x_i) \quad (5.1)$$

Os valores das sinápses, y_i , são determinados a partir de um conjunto de regras *fuzzy* do tipo *se-então*. Considerando que o universo de discurso da entrada x_i é dividido em “ p ” partições *fuzzy*, descritas por funções de pertinência mostradas na figura 5.2, tem-se as seguintes p regras:

R_1^1 Se x_i é A_{i1} Então y_i é w_{i1}

R_1^2 Se x_i é A_{i2} Então y_i é w_{i2}

⋮

R_1^p Se x_i é A_{ip} Então y_i é w_{ip}

Para este conjunto de regras, as variáveis x_i e y_i são determinísticas e A_{ij} são conjuntos *fuzzy*, definidos por $\mu_{ij}(x_i)$. Isto exige as etapas de *fuzzificação* para x_i e *defuzzificação* para determinar as sinápses y_i . Considerando funções de pertinência complementares, figura 5.2, somente para no máximo duas partições o valor da função de pertinência é diferente de zero. Isto equivale a dizer que, para cada variável de entrada x_i , no máximo duas das “ p ” regras são ativas, sendo indexadas por k_i e k_i+1 . Com isto, utilizando o esquema de inferência *max-min*, mostrado na figura 5.3, e o centro de gravidade para efetuar a *defuzzificação*, os valores das sinápses podem ser determinados da seguinte forma:

$$f_i(x_i) = \frac{\sum_{j=1}^p \mu_{ij}(x_i) \cdot w_{ij}}{\sum_{j=1}^p \mu_{ij}(x_i)} = \frac{\mu_{ik_i}(x_i) \cdot w_{ik_i} + \mu_{ik_i+1}(x_i) \cdot w_{ik_i+1}}{\mu_{ik_i}(x_i) + \mu_{ik_i+1}(x_i)} \quad (5.2)$$

Como as funções de pertinência são complementares, a soma de duas funções sucessivas é igual a 1. Portanto, tem-se a expressão final para a sinápsis:

$$f_i(x_i) = \mu_{ik_i}(x_i) \cdot w_{ik_i} + \mu_{ik_i+1}(x_i) \cdot w_{ik_i+1} \quad (5.3)$$

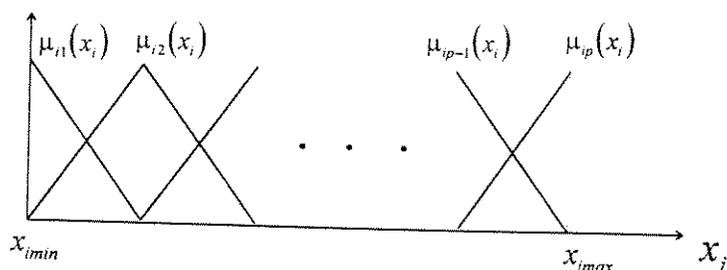


Fig. 5.2 - Funções de pertinência

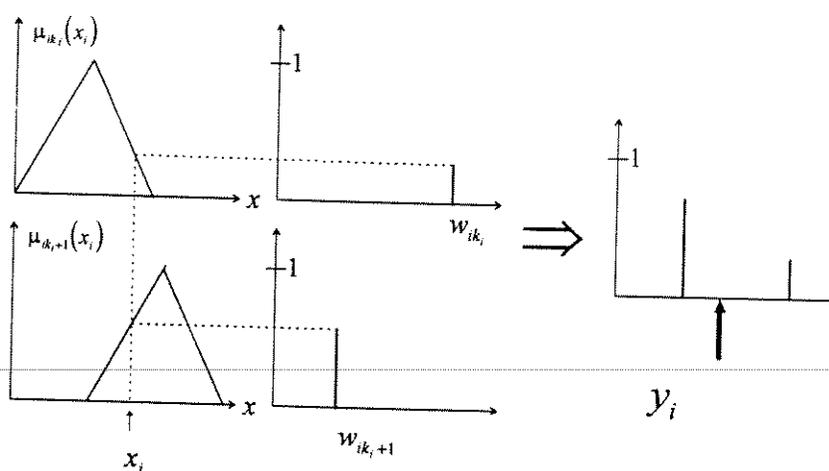


Fig. 5.3 - Mecanismo de Inferência do NFN

Como visto acima, somente ramos ativados são relevantes para o processamento da rede. Assim, só um ou dois pesos correspondentes aos ramos ativados são ajustados durante o treinamento do NFN. Este mecanismo sugere que um NFN deve exibir período de treinamento muito menor do que redes neurais convencionais (Nunes e Gomide, 1996).

5.2.3 - APRENDIZADO DO NFN

De forma similar à rede neural, o método de aprendizagem do NFN consiste em ajustes de seus parâmetros e funções. Será adotado o aprendizado supervisionado, onde somente serão ajustados os parâmetros w_{ij} , consequentes das regras *fuzzy*, que daqui em diante serão chamados de pesos. Isto porque o ajuste dos valores modais das funções de pertinência implicaria em minimização de funções com mínimos locais, podendo comprometer a convergência do algoritmo de treinamento (Uchino *et alii*, 1995). Essa técnica de aprendizado consiste na apresentação de uma seqüência de entradas (padrões de treinamento) ao NFN, que uma vez processada a informação, fornece o valor da saída. Essa saída é comparada com

o valor desejado, gerando o erro, E , que é usado para ajustar os pesos do NFN, conforme figura 5.4. Estes passos são normalmente repetidos várias vezes até que o NFN tenha “aprendido”, ou seja, que ele seja capaz de responder corretamente, segundo algum critério, a determinadas entradas a ele fornecidas (Nunes e Gomide, 1996). A seguir é discutida uma forma de atualização automática dos pesos do NFN.

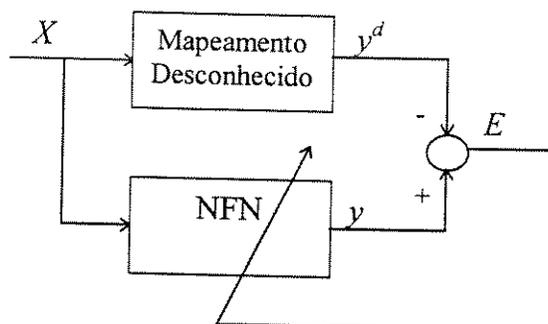


Fig. 5.4 - Processo de ajuste dos parâmetros do NFN

5.2.3.1 - Atualização dos Pesos com Taxa de Aprendizagem Ótima

O procedimento de atualização dos pesos descrito a seguir consiste de atualização incremental. Outros procedimentos, tais como, atualização em lote e atualização incremental com termo de *Momentum*, foram apresentados em (Yamakawa *et alii*, 1992) e Nunes e Gomide (1996). Vale ressaltar que, para esses dois, não se verificou uma grande vantagem com relação ao tempo de processamento e erro de aproximação, sobre a atualização incremental, conforme mostrado em (Yamakawa *et alii*, 1992).

Seja $X_t = (x_{t1}, x_{t2}, \dots, x_{ti}, \dots, x_{tm})$ o t -ésimo padrão de entrada aplicado ao NFN; y_t a saída calculada e y_t^d seu respectivo valor desejado, onde $t=1, 2, \dots, t_{max}$, e t_{max} o número de padrões. Com os valores de y_t e y_t^d pode-se calcular o erro quadrático definido por:

$$E_t = \frac{1}{2}(y_t - y_t^d)^2 = E_t(w_{ij}) \quad (5.4)$$

O algoritmo de treinamento consiste na atualização dos pesos para cada padrão de entrada, de forma a minimizar o erro E_t :

$$P_1 \quad \begin{cases} \min E_t = E_t(w_{ij}) \\ w_{ij} \in \mathfrak{R} \\ \forall i = 1, \dots, n; j = 1, \dots, p \end{cases} \quad (5.5)$$

Para cada coordenada de entrada, do padrão apresentado ao NFN, somente as funções $\mu_{ik_i}(x_{ij})$ e $\mu_{ik_i+1}(x_{ij})$ são diferentes de zero. Isto permite reescrever o problema P_1 da seguinte forma:

$$P_2 \quad \boxed{\begin{array}{l} \min E_i = E_i(w_{ij}) = E_i(\tilde{w}) \\ w_{ij} \in \mathfrak{R} \\ \forall i = 1, \dots, n; j = k_i, (k_i + 1) \end{array}} \quad (5.6)$$

$$\text{sendo } \tilde{w} = \left[w_{1k_1} \ w_{1(k_1+1)} \ \dots \ w_{ik_i} \ w_{i(k_i+1)} \ \dots \ w_{nk_n} \ w_{n(k_n+1)} \right].$$

A vantagem de resolver P_2 ao invés de P_1 é que a função a ser minimizada em P_2 possui "2n" variáveis, não dependendo do número de partições *fuzzy*, ao passo que em P_1 o número de variáveis é "np", que é maior ou igual a "2n". Como foi dito anteriormente, a função a ser minimizada $E_i(\tilde{w})$ é quadrática e convexa. Para provar a convexidade da função basta mostrar que a matriz hessiana, H , é semidefinida positiva. De (5.3) e (5.4) tem-se:

$$\nabla E_i(\tilde{w}) = (y_i - y_i^d) \begin{bmatrix} \mu_{1k_i} \\ \mu_{1k_i+1} \\ \vdots \\ \mu_{ik_i} \\ \vdots \\ \mu_{nk_n+1} \end{bmatrix} \quad (5.7)$$

$$\therefore H = \begin{bmatrix} \mu_{1k_i}^2 & \mu_{1k_i} \cdot \mu_{1k_i+1} & \dots & \mu_{1k_i} \cdot \mu_{ik_i} & \dots & \mu_{1k_i} \cdot \mu_{nk_n+1} \\ \mu_{1k_i} \cdot \mu_{1k_i+1} & \mu_{1k_i+1}^2 & \dots & \dots & \dots & \mu_{1k_i+1} \cdot \mu_{nk_n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mu_{1k_i} \cdot \mu_{ik_i} & \dots & \dots & \mu_{ik_i}^2 & \dots & \mu_{ik_i} \cdot \mu_{nk_n+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{1k_i} \cdot \mu_{nk_n+1} & \mu_{1k_i+1} \cdot \mu_{nk_n+1} & \dots & \mu_{ik_i} \cdot \mu_{nk_n+1} & \dots & \mu_{nk_n+1}^2 \end{bmatrix} \quad (5.8)$$

onde $\nabla E_i(\tilde{w})$ é o vetor gradiente de $E_i(\tilde{w})$ com relação a \tilde{w} .

Seja $V = [v_1 \ v_2 \ \dots \ v_{2n}]^T \in \mathfrak{R}^{2n}$ um vetor qualquer. A expressão $V^T \cdot H \cdot V$, que permite avaliar se uma matriz é definida positiva, pode ser obtida por indução matemática, fazendo $n=1, 2, 3, \dots$, obtendo-se:

$$V^T \cdot H \cdot V = \left(v_1 \cdot \mu_{1k_i} + v_2 \cdot \mu_{1k_i+1} + \dots + v_{2n} \cdot \mu_{nk_n+1} \right)^2 \geq 0$$

Portanto, a matriz H é semidefinida positiva e conseqüentemente, $E_t(\tilde{w})$ é convexa. Como o espaço onde $E_t(\tilde{w})$ está definida, \mathbb{R}^{2n} , é convexo, o ponto de mínimo local também é mínimo global (Bazarra *et alii*, 1993).

A função $E_t(\tilde{w})$ é não-linear, portanto, P_2 é um Problema de Programação Não-linear. A programação não-linear caracteriza-se por não possuir um método geral de resolução dos seus problemas. São muitos algoritmos e quase sempre voltados para problemas específicos, prendendo-se a características diversas tais como continuidade, diferenciabilidade de primeira e/ou de segunda ordem, e outras. Uma outra particularidade é a inexistência de critérios absolutos para comparação entre os algoritmos existentes (Mateus e Luna, 1986).

A estrutura geral de um algoritmo de minimização irrestrita, que é o caso de P_2 , é:

Passo 1: Inicialização

escolha um ponto $\tilde{w}^0 \in \mathbb{R}^d$,
 $j \leftarrow 0$;

Passo 2: Se \tilde{w}^j é o ponto desejável ($E_t(\tilde{w}^j) \leq \varepsilon$)

então vá para o passo 4
 senão vá para o passo 3

Passo 3: Escolher a direção e executar busca

determine, por regra fixa, uma direção de \tilde{h}^j e execute um processo de busca nesta direção, determinando \tilde{w}^{j+1} , onde:
 $\tilde{w}^{j+1} = \tilde{w}^j + \alpha^j \cdot \tilde{h}^j$, tal que $E_t(\tilde{w}^{j+1}) < E_t(\tilde{w}^j)$

$j \leftarrow j+1$;
 volte ao passo 2

Passo 4: fim

• **Detalhamento do Algoritmo**

Passo 1: Inicialização

A escolha do ponto inicial é importante pois influencia consideravelmente o desempenho do método. Para a NFN a condição inicial será nula, sem comprometer a convergência do algoritmo.

Passo 3a: Determinação da Direção de Busca

A escolha das direções caracteriza um algoritmo de minimização irrestrita. O método mais simples é o do *Gradiente*, que será adotado aqui. Para este método a direção é determinada da seguinte forma:

$$\tilde{h}^j = -\nabla E_t(\tilde{w}^j) \tag{5.9}$$

sendo $\nabla E_t(\tilde{w}^j)$ o vetor gradiente de $E_t(\tilde{w}^j)$.

Portanto, uma componente w_{ik}^j é atualizada da seguinte forma:

$$w_{ik}^{j+1} = w_{ik}^j - \alpha^j \cdot \frac{\partial E_t(w_{ik}^j)}{\partial w_{ik}^j} \quad (5.10)$$

$$\frac{\partial E_t}{\partial w_{ik}^j} = (y_t^j - y_t^d) \cdot \frac{\partial y_t^j}{\partial w_{ik}^j} \quad (5.11)$$

$$\frac{\partial y_t^j}{\partial w_{ik}^j} = \mu_{ik}(x_{ti}) \quad (5.12)$$

$$\frac{\partial E_t}{\partial w_{ik}^j} = (y_t^j - y_t^d) \cdot \mu_{ik}(x_{ti}) \quad (5.13)$$

$$\therefore w_{ik}^{j+1} = w_{ik}^j - \alpha^j \cdot (y_t^j - y_t^d) \cdot \mu_{ik}(x_{ti}) \quad (5.14)$$

Passo 3b: Determinação do Tamanho do passo

O tamanho do passo, termo α , que no caso de treinamento de redes neurais ou *neurofuzzy* é chamado de taxa de aprendizado, pode ser determinado empiricamente, ou utilizando algum método de busca unidirecional indireto (por exemplo Seção Áurea, Fibonacci e outros) ou direto (fazendo aproximação polinomial para a função a ser minimizada) (Bazaraa *et alii*, 1993; Mateus e Luna, 1986). Em Nunes e Gomide (1996) é utilizado um método, empregado em treinamento de redes neurais, para modificação dinâmica da taxa de aprendizado. Esse método, no entanto, não utiliza uma propriedade particular do NFN, que é o fato da função objetivo a ser minimizada ser quadrática. Levando isto em consideração é proposta uma alternativa para obtenção da taxa de aprendizado a partir do resultado do teorema 5.1.

Teorema 5.1: Dado um padrão $X_t = (x_{t1}, x_{t2}, \dots, x_{tn}, \dots, x_{tm})$ e y_t^d , valor desejado para o mapeamento $y = f(X)$, é possível determinar uma expressão fechada para a taxa de aprendizado α , tal que, utilizando o método do gradiente, os pesos do NFN que proporcionam erro de aproximação nulo pode ser determinado em um passo de cálculo.

Prova:

Dado \tilde{w}^0 o objetivo é obter α tal que o erro:

$$E_i^l = \frac{1}{2}(y_i^l - y_i^d)^2 = \frac{1}{2} \left(\left(\sum_{i=1}^n w_{ik_i}^1 \cdot \mu_{k_i}(x_i) + w_{ik_i+1}^1 \cdot \mu_{k_i+1}(x_i) \right) - y_i^d \right)^2 = 0 \quad (5.15)$$

para

$$w_{ik_i}^1 = w_{ik_i}^0 - \alpha \cdot \left[\sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] - y_i^d \right] \cdot \mu_{k_i}(x_{ii}) \quad (5.16)$$

$$w_{ik_i+1}^1 = w_{ik_i+1}^0 - \alpha \cdot \left[\sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] - y_i^d \right] \cdot \mu_{k_i+1}(x_{ii}) \quad (5.17)$$

A equação (5.15) é satisfeita se:

$$\sum_{i=1}^n w_{ik_i}^1 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^1 \cdot \mu_{k_i+1}(x_{ii}) = y_i^d \quad (5.18)$$

Multiplicando os dois lados da equação (5.16) por $\mu_{k_i}(x_i)$ e (5.17) por $\mu_{k_i+1}(x_i)$, obtém-se:

$$w_{ik_i}^1 \cdot \mu_{k_i}(x_{ii}) = w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) - \alpha \cdot \left[\sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] - y_i^d \right] \cdot \left[\mu_{k_i}(x_{ii}) \right]^2 \quad (5.19)$$

$$w_{ik_i+1}^1 \cdot \mu_{k_i+1}(x_{ii}) = w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) - \alpha \cdot \left[\sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] - y_i^d \right] \cdot \left[\mu_{k_i+1}(x_{ii}) \right]^2 \quad (5.20)$$

para $i=1,2,\dots,n$.

Somando as “2n” equações, tem-se:

$$\begin{aligned} \sum_{i=1}^n \left[w_{ik_i}^1 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^1 \cdot \mu_{k_i+1}(x_{ii}) \right] &= \sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_{ii}) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] \\ &\quad - \alpha \cdot \left[\sum_{i=1}^n \left[w_{ik_i}^0 \cdot \mu_{k_i}(x_i) + w_{ik_i+1}^0 \cdot \mu_{k_i+1}(x_{ii}) \right] - y_i^d \right] \cdot \left[\sum_{i=1}^n \mu_{k_i}(x_{ii})^2 + \mu_{k_i+1}(x_{ii})^2 \right] \end{aligned} \quad (5.21)$$

de (5.18) tem-se que o lado esquerdo da equação é igual a y_i^d ; portanto:

$$\alpha \cdot \left[\sum_{i=1}^n \left[w_{ki}^0 \cdot \mu_{ik}(x_{ii}) + w_{ki+1}^0 \cdot \mu_{ik+1}(x_{ii}) \right] - y_i^d \right] \cdot \left[\sum_{i=1}^n \mu_{ik}(x_{ii})^2 + \mu_{ik+1}(x_{ii})^2 \right] = \left[\sum_{i=1}^n \left[w_{ki}^0 \cdot \mu_{ik}(x_{ii}) + w_{ki+1}^0 \cdot \mu_{ik+1}(x_{ii}) \right] - y_i^d \right] \quad (5.22)$$

Da equação (5.22) tem-se:

$$\alpha \cdot \left[\sum_{i=1}^n \mu_{ik}(x_{ii})^2 + \mu_{ik+1}(x_{ii})^2 \right] = 1 \quad (2.23)$$

$$\therefore \alpha = \frac{1}{\sum_{i=1}^n \mu_{ik}(x_{ii})^2 + \mu_{ik+1}(x_{ii})^2} \quad (5.24)$$

O resultado deste teorema anterior diz que, independente da condição inicial \tilde{w}^0 , é possível encontrar \tilde{w}^1 tal que o erro seja nulo para o padrão "r". Este resultado será usado no capítulo 6, em problema de detecção de falhas empregando a técnica baseada em identificação de parâmetros do modelo

5.3 - NEO FUZZY NEURON GENERALIZADO (NFNG)

5.3.1 - ESTRUTURA DO NFNG

A estrutura do Neo Fuzzy Neuron, apresentada no item anterior, detalhada em (Yamakawa *et alii*, 1992; Uchino *et alii*, 1995; Nunes e Gomide, 1996), permite um mapeamento não-linear $X \rightarrow Y$, onde $X \subset \mathcal{R}^n$ e $Y \subset \mathcal{R}$, ou seja, o número de saídas é igual a um. Para o caso de sistema com "m" saídas, poderiam ser utilizados "m" NFN independentes. Uma outra opção é utilizar a estrutura em forma de rede, chamada de *Neo Fuzzy Neuron-Generalizado* (NFNG), mostrada na figura 5.5. Usando esta configuração a etapa de *fuzzificação* é feita de uma só vez, reduzindo assim o tempo de processamento.

De forma análoga ao NFN, para a NFNG as saídas, y_l , e as sinápses, y_{li} , são calculadas da seguinte forma:

$$y_l = \sum_{i=1}^n y_{li}(x_i) = \sum_{i=1}^n f_{li}(x_i) = f_{l1}(x_1) + f_{l2}(x_2) + \dots + f_{ln}(x_n) \quad (5.25)$$

$$f_{li}(x_i) = \mu_{ik_i}(x_i) \cdot w_{ik_i,l} + \mu_{ik_i+1}(x_i) \cdot w_{i(k_i+1)l} \quad (5.26)$$

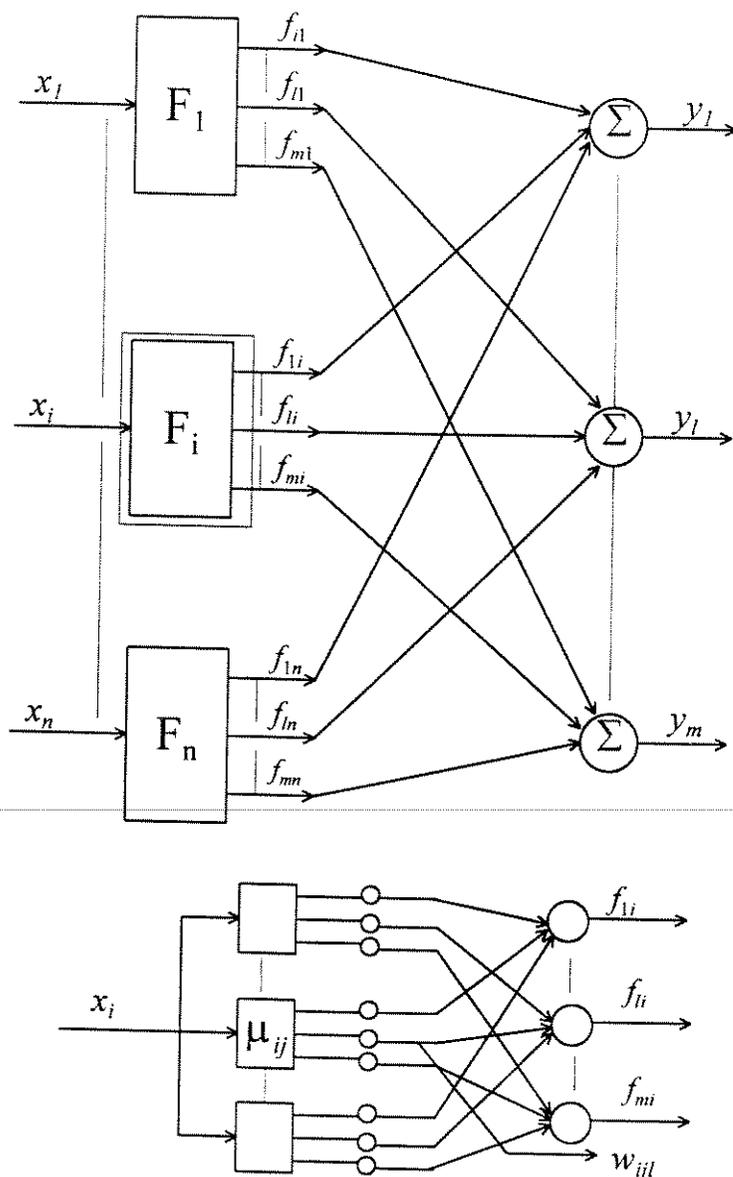


Fig. 5.5 - Estrutura do Neo-Fuzzy-Neuron-Generalizado

5.3.2 - TREINAMENTO DO NFNG

O treinamento do NFNG consiste em minimizar, para cada padrão apresentado, os m erros, definidos por:

$$E_{il} = \frac{1}{2} (y_{il} - y_{il}^d)^2 = E_t(w_{ijl}), \text{ para } l = 1, 2, \dots, m \quad (5.27)$$

Portanto, para ajustar os pesos do NFNG são resolvidos “ m ” problemas tipo P_2 , para cada padrão apresentado.

5.3.3 - GERAÇÃO DAS FUNÇÕES DE PERTINÊNCIA DO NFNG

Em alguns testes realizados com o NFN foi verificado que o aumento do número de partições *fuzzy* nem sempre é acompanhado de uma melhor precisão. Além disso, quando existir diferença de comportamento dinâmico (variáveis com comportamento próximo de linear e fortemente não linear), pode ser interessante utilizar números diferentes de partições *fuzzy* para as entradas do NFN. Por isso, as funções de pertinência para o NFNG são obtidas utilizando-se o algoritmo de *clusterização*, baseado em rede neural auto-organizada descrita no capítulo 2. Como foi visto, os centros dos *clusters* correspondem aos valores modais, a_{ij} , das funções de pertinência $\mu_{ij}(x_i)$, figura 5.6. O algoritmo de *clusterização*, além de fornecer os valores a_{ij} , fornece o número de partições *fuzzy* para cada entrada, p_i .

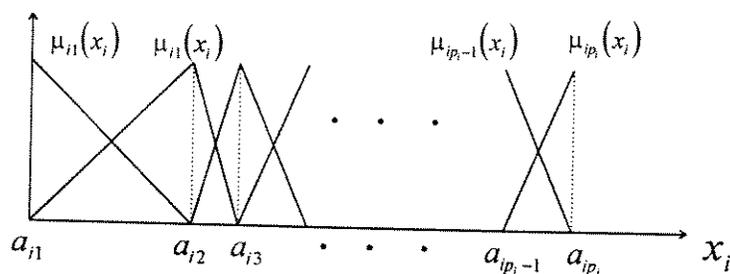


Fig. 5.6 - Funções de pertinência para o NFNG

Para que o processamento do NFNG seja feito com número reduzido de operações, as funções de pertinência são caracterizadas por três parâmetros (a_{ij}, α_{eij} e α_{dij}), onde:

$$\alpha_{eij} = \frac{1}{a_{ij} - a_{ij-1}} \text{ e } \alpha_{dij} = \frac{1}{a_{ij} - a_{ij+1}}$$

são as derivadas da função de pertinência à esquerda e direita de a_{ij} , respectivamente. Com estes parâmetros o cálculo das funções de pertinência é bastante simples, envolvendo duas operações de soma e uma de multiplicação.

$$\mu_{ij}(x_i) = \begin{cases} \alpha_{eij} \cdot (x_i - a_{ij-1}) + 1, & a_{ij-1} \leq x_i \leq a_{ij} \\ \alpha_{dij} \cdot (a_{ij+1} - x_i) + 1, & a_{ij} < x_i \leq a_{ij+1} \\ 0 & \text{caso contrário} \end{cases} \quad (5.28)$$

5.4 - IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS

O *neo-fuzzy-neuron-generalizado* (NFNG) apresentado no item anterior é uma estrutura de rede não-recorrente. Devido ao fato deste tipo de estrutura não apresentar memória dinâmica, a sua utilização para representar sistemas dinâmicos só é possível com a inclusão, no vetor de entradas da rede, de entradas e saídas do sistema em instantes anteriores de tempo ($k-1$, $k-2$, ..., $k-M$), conforme apresentado na figura 5.7. Este vetor de entradas é conhecido como linha de derivação de atraso (LDA) e transforma um problema de modelagem temporal (aprendizagem do comportamento dinâmico de um sistema no domínio do tempo) num problema de modelagem espacial (mapeamento estático das entradas e saídas, juntamente com seus respectivos retardos no tempo, na próxima saída) (Zuben, 1993).

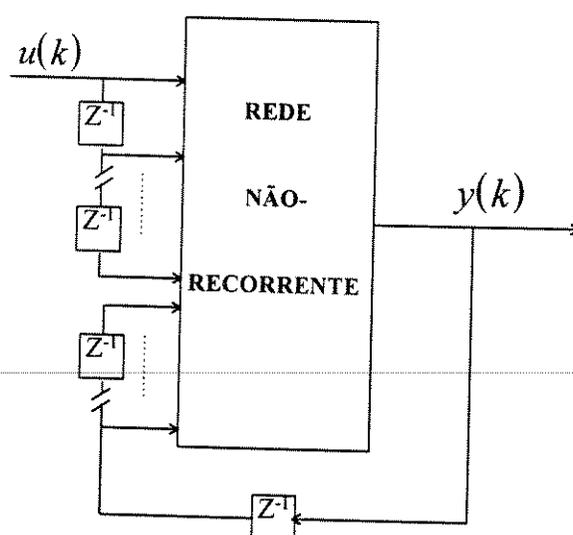


Fig. 5.7 - Representação de comportamento dinâmico utilizando rede *neurofuzzy* não-recorrente

A utilização da LDA apresenta como principal desvantagem o aumento do espaço de entrada da rede. Mas, no caso do NFNG este aumento não chega a preocupar, pois o seu processamento é bastante rápido, uma vez que somente ramos ativados são relevantes para calcular as saídas da rede e os problemas tratados neste trabalho são de pequena dimensão.

Uma outra forma de representar a dinâmica do sistema é utilizar uma rede recorrente. Nessa estrutura, as conexões recorrentes provêm a rede a possibilidade de construir representações dinâmicas internas, o que vai permitir a existência de memória dinâmica (Zuben, 1993; Nunes e Gomide, 1996). Por outro lado o algoritmo de treinamento das redes recorrentes é mais complexo e exige mais cálculos.

5.4.1 - ESTRATÉGIA DE TREINAMENTO DO NFNG PARA IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS

Para facilitar a notação, considere o sistema dinâmico discreto com uma entrada e uma saída, representado por:

$$y(k+1) = f(y(k), \dots, y(k-r+1); u(k), \dots, u(k-q+1)) \quad (5.29)$$

onde f é uma função não conhecida que se pretende identificar; u é a entrada; y a saída do sistema e r e q duas constantes inteiras.

No treinamento do NFNG, para identificação de sistemas dinâmicos, é adotada a mesma estratégia empregada no treinamento de redes neurais (Narendra e Parthasarathy, 1990), figura 5.8. Durante o treinamento as saídas desejadas, atual e atrasadas no tempo, são usadas pela rede para calcular a próxima saída, ou seja:

$$\hat{y}(k+1) = \hat{f}(y(k), \dots, y(k-r+1); u(k), \dots, u(k-q+1)) \quad (5.30)$$

onde \hat{f} é a função determinada pelo NFNG.

Após o treinamento a saída do NFNG é realimentada, obtendo-se:

$$\hat{y}(k+1) = \hat{f}(\hat{y}(k), \dots, \hat{y}(k-r+1); u(k), \dots, u(k-q+1)) \quad (5.31)$$

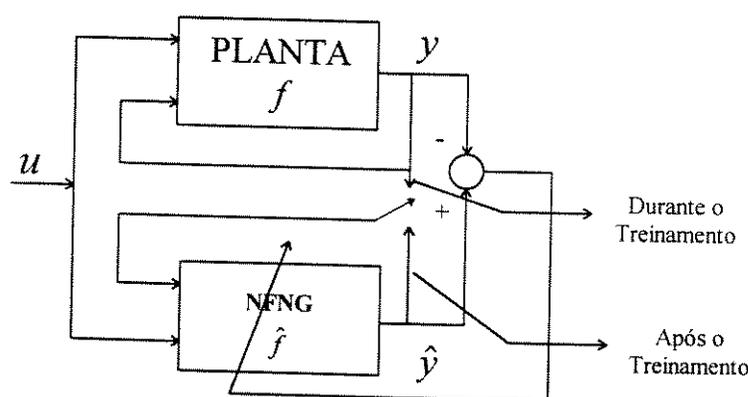


Fig. 5.8 - Diagrama da Estratégia de treinamento do NFNG

5.4.2 - APLICAÇÃO DO NFNG EM IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS

Serão utilizados três exemplos de sistemas dinâmicos apresentados em Narendra e Parthasarathy (1990) e Wang (1994) para testar o desempenho do NFNG. Isto permite fazer uma comparação entre as três estruturas. Além desses três exemplos, é feita a identificação do sistema de acionamento C.C., descrito no capítulo 2.

5.4.2.1 - Exemplo 1

- **Modelo do Sistema**

O sistema dinâmico deste exemplo é definido pela equação a diferenças:

$$y(k+1) = \alpha_1 \cdot y(k) + \alpha_2 \cdot y(k-1) + g[u(k)] \quad (5.32)$$

Este modelo possui duas parcelas: uma linear com relação às saídas atrasadas no tempo, definidas pelas constantes α_1 e α_2 , sendo $\alpha_1=0.3$ e $\alpha_2 = 0.6$, e outra não linear com relação à entrada do sistema, definida por:

$$g(u) = 0.6 \text{sen}(\pi u) + 0.3 \text{sen}(3\pi u) + 0.1 \text{sen}(5\pi u)$$

sendo que $g(u)$ é a função que se pretende identificar.

Portanto, o que se pretende é obter um conjunto de pesos de um NFNG, tal que:

$$\hat{y}(k+1) = \alpha_1 \cdot \hat{y}(k) + \alpha_2 \cdot \hat{y}(k-1) + \hat{g}[u(k)] \quad (5.33)$$

seja uma boa aproximação para $y(k+1)$.

- **Estrutura do NFNG**

Como a função que se pretende identificar depende de $u(k)$ somente, o NFNG possui uma entrada, o próprio $u(k)$, e uma saída, $\hat{g}(u(k))$. Foram feitos testes com diversas partições *fuzzy* da entrada, mas vinte foi número o que apresentou melhor resultado. As funções de pertinência são do tipo mostradas na figura 5.6.

- **Metodologia de Treinamento do NFNG**

Para o treinamento do NFNG, o valor de $u(k)$ foi gerado aleatoriamente, com distribuição de probabilidade uniforme no intervalo $[-1,1]$, conforme figura 5.9. Esta política de treinamento permite avaliar a capacidade de generalização do mapeamento do NFNG com relação ao tipo de entrada do sistema. Na fase de ajuste dos pesos, foram apresentados dez mil padrões ao NNFG, divididos em dez épocas, com mil padrões cada. Ao final de cada época, com os pesos mantidos constantes, foram apresentados setecentos padrões ao NFNG e calculado o erro quadrático médio, para uma entrada $u(k)$ definida pela equação:

$$u(k) = \text{sen}(2 \cdot \pi \cdot k / 250) \quad (5.34)$$

Na figura 5.10 é mostrada a evolução do erro em função da época, para um NFNG com vinte partições *fuzzy*. Observa-se que a partir da primeira época o erro é inferior a 10^{-3} .

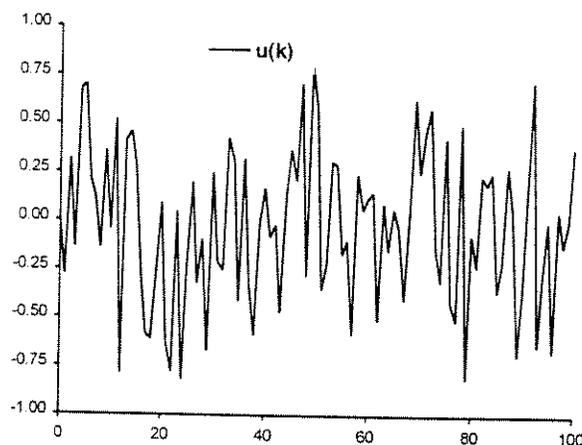


Fig. 5.9 – Sinal de entrada utilizado no treinamento do NFNG

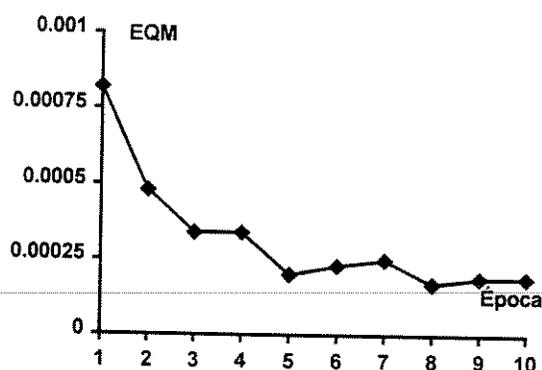


Fig. 5.10 - Erro quadrático médio em função da época de treinamento.

• Resultados Obtidos com o NFNG

As figuras 5.11a e 5.11b mostram as curvas de $g(u(k))$ e $\hat{g}(u(k))$, e das saídas desejada e estimada, respectivamente. O resultado apresentado é para uma entrada $u(k)$ definida pela equação (5.34). Já na figura 5.12 é mostrado o resultado da identificação para a entrada $u(k)$ definida por:

$$u(k) = \begin{cases} \text{sen}(2\pi.k/250), & 251 \leq k \leq 500 \\ 0.5 \text{sen}(2\pi.k/25) + 0.5 \text{sen}(2\pi.k/250), & \text{caso contrário} \end{cases} \quad (5.35)$$

A figura 5.13 mostra a influência do número de partições *fuzzy*. Nota-se que o comportamento do erro quadrático médio não é função estritamente decrescente com o aumento do número de partições. Portanto, uma política que pode ser adotada é começar com número de partições pequeno e à medida que este número for aumentado verificar o comportamento do erro. Para este exemplo o número de partições que proporcionou o menor erro foi igual a vinte.

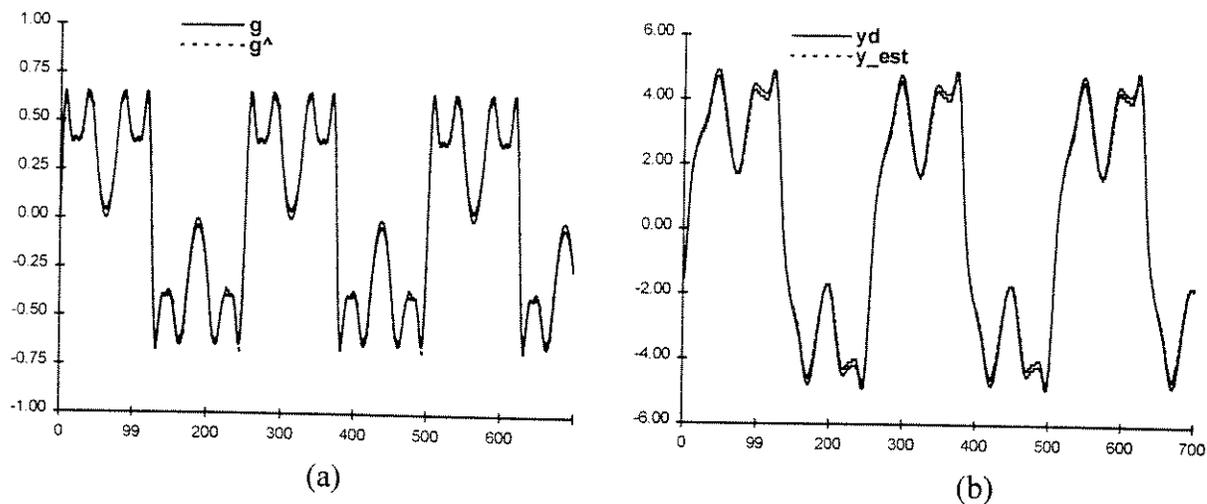


Fig. 5.11 – Resultado da identificação para o exemplo 1 usando o NFNG.
Com $u(k)$ definida por (5.34).

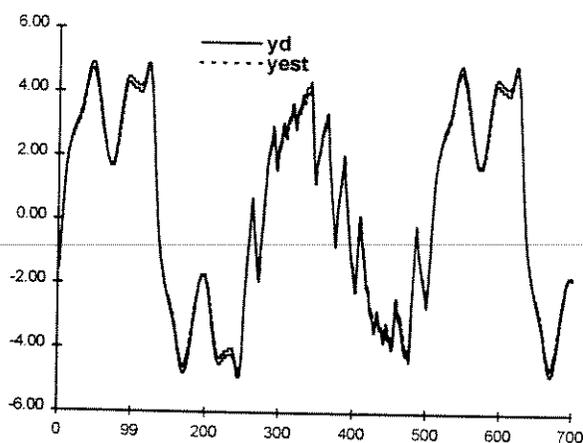


Fig. 5.12 - Saídas real e identificada para o exemplo 1 usando o NFNG.
Com $u(k)$ definida por (5.35).

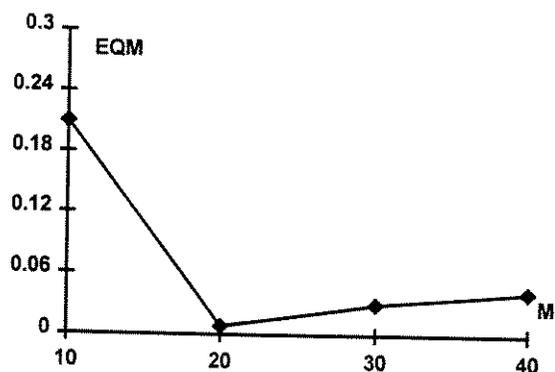


Fig. 5.13 - Erro quadrático médio em função do número de partições *fuzzy*.
Após o treinamento e para entrada $u(k)$ definida por (5.34).

No apêndice B é mostrada a equação completa do modelo identificado, contendo os pesos do NFNG, para o número de partições igual a vinte.

- **Resultados Obtidos por Wang (1994)**

Para efeito de comparação é apresentado o resultado obtido por Wang (1994) usando sistema *fuzzy* adaptativo com 40 regras, implementado por uma rede *neurofuzzy*, figura 5.14. Comparando as figuras 5.12 e 5.14, verifica-se que a rede do Wang proporciona um resultado melhor do que o NFNG.

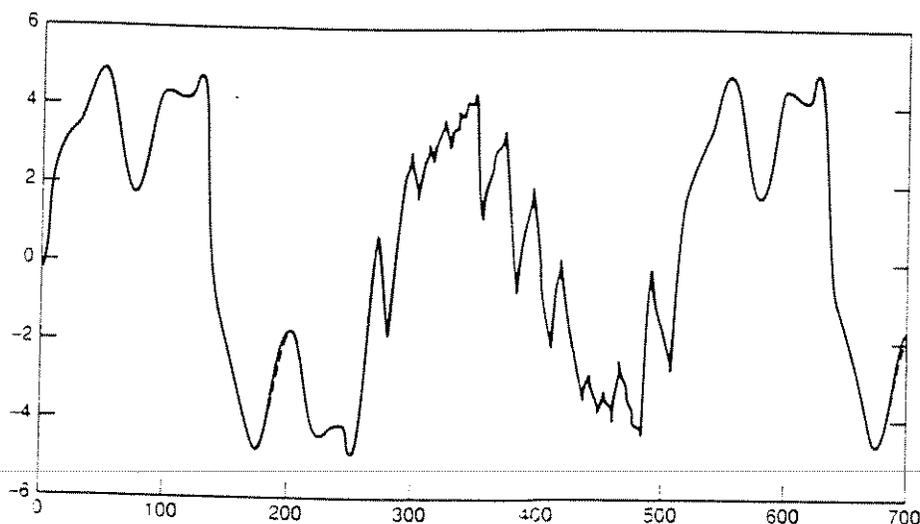


Fig. 5.14 - Saídas real (—) e identificada (---) para o exemplo 1 obtido por Wang (1994), usando 40 funções de pertinência gaussianas. Com $u(k)$ definida por (5.35).

5.4.2.- Exemplo 2

- **Modelo do Sistema**

O sistema dinâmico deste exemplo é não linear com relação às saídas atrasadas e linear com relação à entrada, sendo governado pela equação a diferenças:

$$y(k+1) = g[y(k), y(k-1)] + u(k) \quad (5.36)$$

onde

$$g[y(k), y(k-1)] = \frac{y(k)y(k-1)[y(k) + 2.5]}{1 + y^2(k) + y^2(k-1)} \quad (5.37)$$

é a incógnita e $u(k) = \text{sen}(2\pi k / 25)$.

$$\therefore \hat{y}(k+1) = \hat{g}[\hat{y}(k), \hat{y}(k-1)] + u(k) \quad (5.38)$$

- **Estrutura do NFNG**

O NFNG usado para identificação deste sistema possui:

- duas entradas $y(k)$ e $y(k-1)$;
- uma saída $\hat{g}[\bullet]$;
- cinco partições *fuzzy* para cada entrada.

- **Metodologia de Treinamento do NFNG**

A metodologia de treinamento foi a mesma adotada no exemplo anterior, só que foram apresentados vinte mil padrões com entrada do sistema $u(k)$ gerada aleatoriamente no intervalo $[-1,1]$.

- **Resultados Obtidos com o NFNG**

A figura 5.15 mostra as saídas real e identificada pela rede para uma entrada $u = \text{sen}(2\pi k / 25)$, após o treinamento.

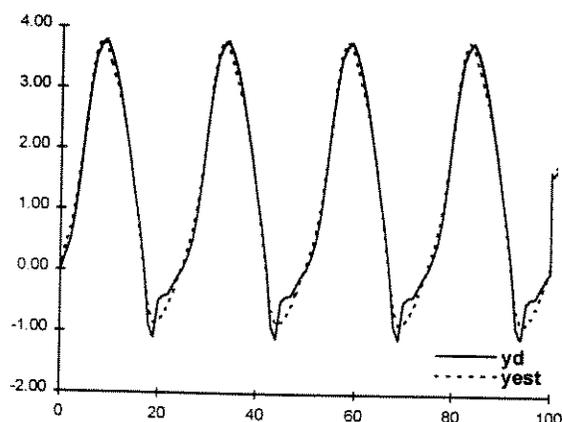


Fig. 5.15 - Saídas real e identificada para o exemplo 2 com NFNG, usando cinco partições *fuzzy* para cada entrada.

- **Resultados Obtidos por Wang (1994) e Narendra e Parthasarathy (1990)**

As figuras 5.16 e 5.17 mostram os resultados obtidos por Wang (1994) e Narendra e Parthasarathy (1990), respectivamente. Foram utilizadas quarenta funções de pertinência gaussianas, no caso do sistema *neurofuzzy* de Wang (1994). A rede neural empregada é de duas camadas ocultas com vinte neurônios na primeira e dez na segunda, respectivamente.

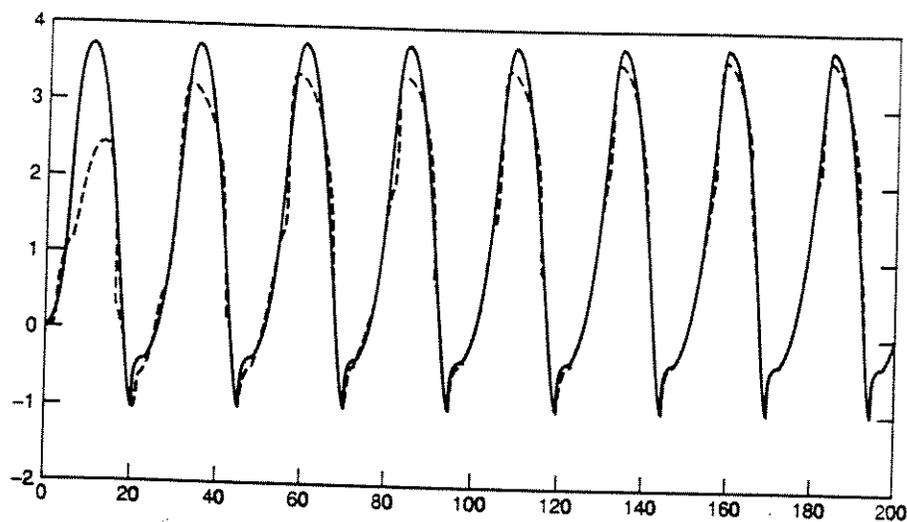


Fig. 5.16 - Saídas real (—) e identificada (---) para o exemplo 2 obtido por Wang (1994), usando 40 funções de pertinência gaussianas para cada entrada.

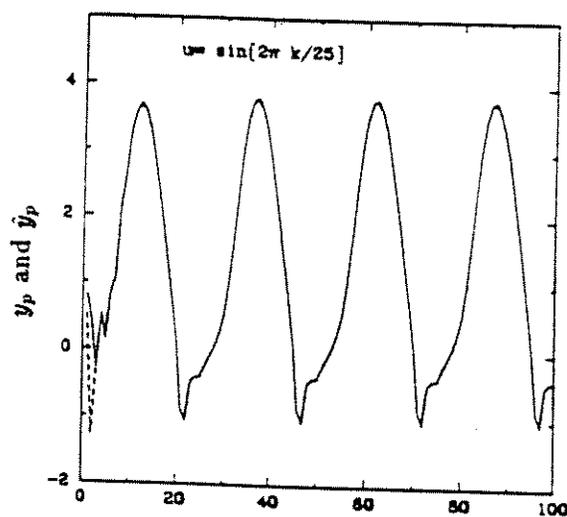


Fig. 5.17 - Saídas real (—) e identificada (---) para o exemplo 2 obtido por Narendra e Parthasarathy (1990), usando uma rede neural com duas camadas ocultas (com 20 e 10 neurônios).

Comparando os gráficos verifica-se que o NNFG apresenta resultado melhor que o da rede de Wang (1994) e pior que da rede neural (Narendra e Parthasarathy, 1990).

5.4.2.3 - Exemplo 3

• Modelo do Sistema

O sistema dinâmico deste exemplo é não-linear com relação as entradas e saídas atuais e atrasadas no tempo, sendo governado pela equação a diferenças:

$$y(k+1) = g[y(k), y(k-1), y(k-2), u(k), u(k-1)] \quad (5.39)$$

onde

$$g(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_4^2} \quad (5.40)$$

$$e \ u(k) = \begin{cases} \text{sen}(2\pi k / 250), & \forall k \leq 500 \\ 0.8 \text{sen}(2\pi k / 250) + 0.2 \text{sen}(2\pi k / 25), & \forall k > 500 \end{cases} \quad (5.41)$$

$$\therefore \hat{y}(k+1) = \hat{g}[\hat{y}(k), \hat{y}(k-1), \hat{y}(k-2), u(k), u(k-1)] \quad (5.42)$$

• Estrutura do NFNG

O NFNG usado para identificação deste sistema possui:

- cinco entradas $y(k)$, $y(k-1)$, $y(k-2)$, $u(k)$ e $u(k-1)$;
- uma saída $y(k+1) = \hat{g}[\bullet]$;
- três partições *fuzzy* para cada entrada.

• Metodologia de Treinamento do NFNG

Foram adotadas três metodologias de treinamento do NFNG para este exemplo, respectivamente:

1. foram utilizados cinco mil padrões com $u(k)$ gerado aleatoriamente no intervalo $[-1, 1]$;
2. foram gerados duzentos e cinquenta padrões com $u(k)$, determinado da seguinte forma:

$$u(k) = \text{sen}(2\pi k / 250) \text{ para } k = 1, 2, \dots, 250 \quad (5.43)$$

Estes duzentos e cinquenta padrões foram repetidos 20 vezes;

3. foram gerados cinco mil padrões com a entrada, $u(k)$, determinada da seguinte forma:

$$u(k) = a_1 \sin(2\pi k / b_1) + (1 - a_1), \text{ para } k = 1, 2, \dots, 5.000 \quad (5.44)$$

sendo a_1 , e b_1 números aleatórios com densidade de distribuição uniforme nos intervalos $[0, 1]$ e $[25, 250]$, respectivamente, alterados a cada 500 iterações.

• Resultados Obtidos com o NFNG

As figuras 5.18, 5.19 e 5.20 apresentam os resultados utilizando o NFNG para as metodologias de treinamento 1, 2 e 3, respectivamente.

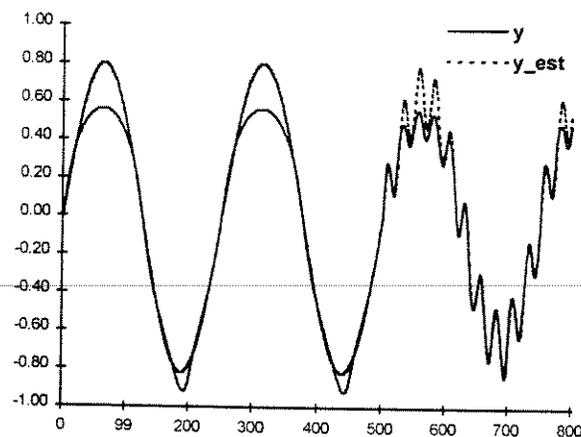


Fig. 5.18 - Saídas real e identificada para o exemplo 3 com NFNG. Metodologia de treinamento 1.

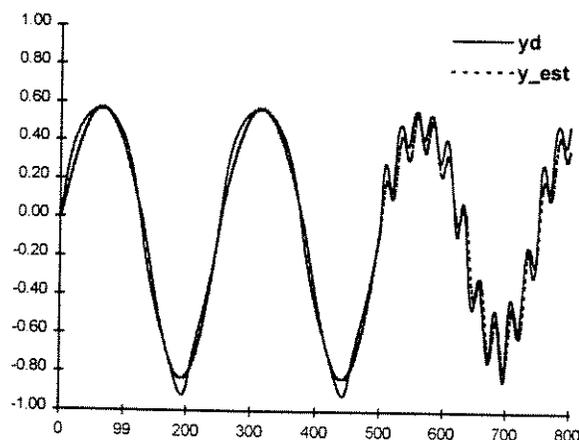


Fig. 5.19 - Saídas real e identificada para o exemplo 3 com NFNG. Metodologia de treinamento 2.

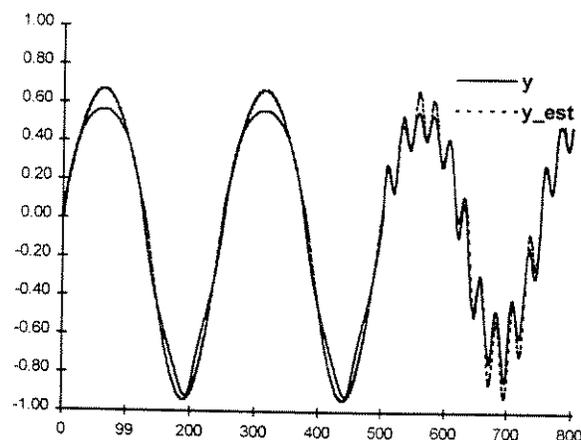


Fig. 5.20 - Saídas real e identificada para o exemplo 3 com NFNG. Metodologia de treinamento 3.

- **Resultados Obtidos por Wang (1994) e Narendra e Parthasarathy (1990)**

As figuras 5.21 e 5.22 mostram os resultados obtidos por Wang (1994) e Narendra e Parthasarathy (1990), respectivamente. Nos dois casos utilizou-se a metodologia 1 de treinamento.

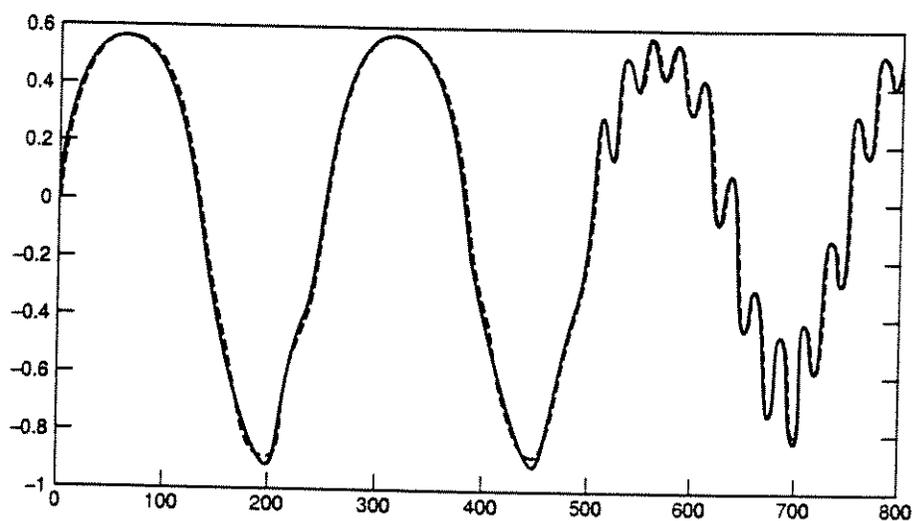


Fig. 5.21 - Saídas real (—) e identificada (---) para o exemplo 3 obtido por Wang (1994), usando 40 funções de pertinência gaussianas para cada entrada.

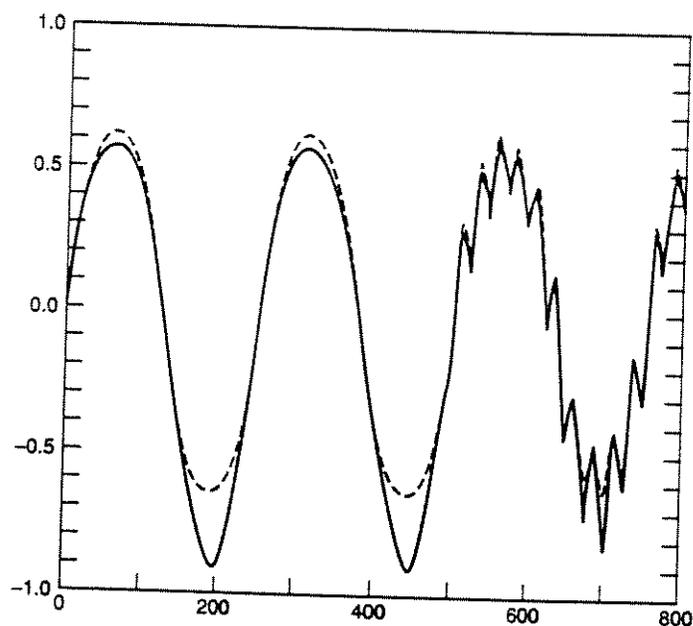


Fig. 5.22 - Saídas real (—) e identificada (---) para o exemplo 3 obtido por Narendra e Parthasarathy (1990), usando uma rede neural com duas camadas ocultas (com 20 e 10 neurônios).

• Influência da Taxa de Aprendizado

Na figura 5.23 é verificada a influência da escolha da taxa de aprendizado na convergência do algoritmo de treinamento do NFNG. Foram feitos treinamentos com a taxa fixa, =0.1 e 0.5 e calculada a cada iteração usando o procedimento descrito anteriormente. É mostrado no gráfico o erro quadrático médio em função da época de treinamento para as três situações. Verifica-se que a escolha de uma taxa grande pode provocar a não convergência do algoritmo. No entanto, a taxa ajustada dinamicamente, utilizando a metodologia proposta, ou fixa num valor adequado possibilita uma convergência rápida. Cada época consiste na apresentação de dois mil e quinhentos padrões, sendo que a entrada do sistema é gerada aleatoriamente.

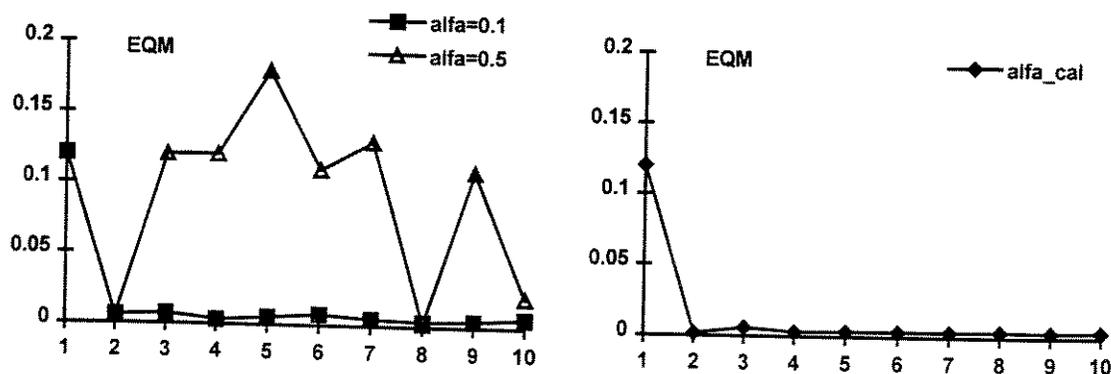


Fig. 5.23 - Erro quadrático médio para identificação usando a metodologia de treinamento 1 em função da época (2.500 padrões) para $\alpha(t)=0.1$; $\alpha(t)=0.5$ e usando a expressão (5.24)

5.4.2.4 - Exemplo 4

- **Modelo do Sistema**

Neste exemplo o sistema dinâmico é o sistema de acionamento C.C., descrito no capítulo 3, representado pelo modelo (3.5).

- **Estrutura do NFNG**

O NFN utilizado possui:

- cinco entradas $y_1(k)$, $y_2(k)$, $y_3(k)$, $u_1(k)$ e $u_2(k)$;
- três saídas $y_1(k+1)$, $y_2(k+1)$, $y_3(k+1)$;
- cinco partições para $y_1(k)$, $y_3(k)$ e $u_1(k)$ e três para $y_2(k)$ e $u_2(k)$;

- **Resultados Obtidos com o NFNG**

As figuras de (5.24) a (5.28) mostram as curvas de saídas identificadas e reais, bem como as entradas após um ciclo de treinamento, com duzentos e cinquenta mil padrões

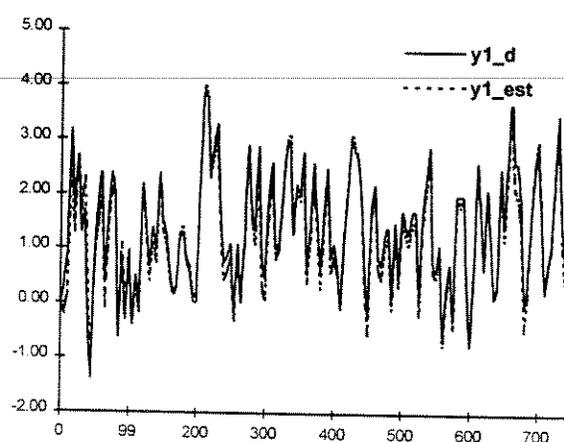


Fig. 5.24 - Saída 1 real e estimada pelo NFNG do exemplo 4

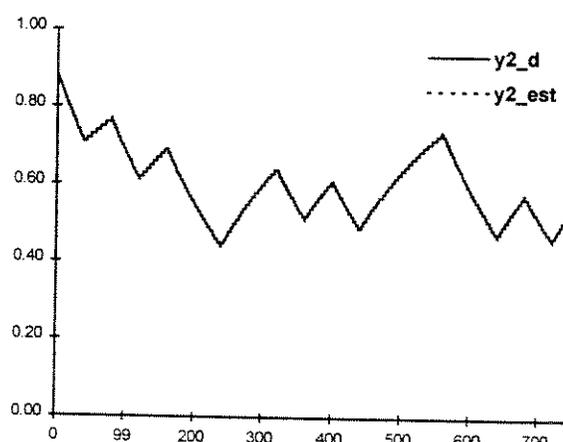


Fig. 5.25 - Saída 2 real e estimada pelo NFNG do exemplo 4

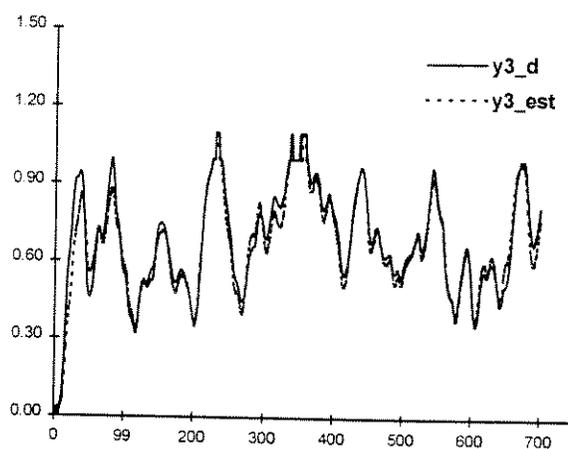


Fig. 5.26 - Saída 3 real e estimada pelo NFN do exemplo 4

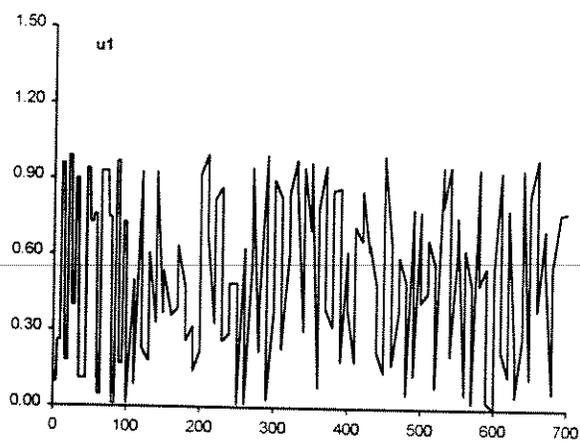


Fig. 5.27 - Entrada 1 do sistema do exemplo 4

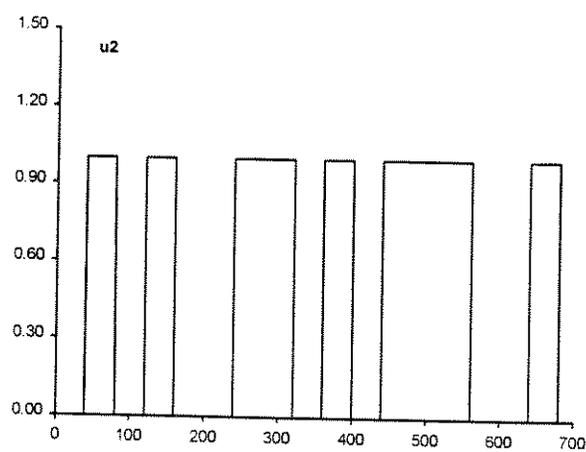


Fig. 5.28 - Entrada 2 do sistema para o exemplo 4

5.5 - ANÁLISE DA COMPLEXIDADE DOS ALGORITMOS: NFNG; REDE NEURAL E REDE NEUROFUZZY DO WANG (1994)

Para avaliar as três estruturas, serão consideradas as expressões que permitem calcular as saídas em função das entradas, que são:

$$\text{NFNG: } y_i = \sum_{i=1}^n \mu_{ik_i}(x_i) \cdot w_{ik_i} + \mu_{ik_i+1}(x_i) \cdot w_{i(k_i+1)} \quad (5.45)$$

$$\text{Wang: } y_i = \frac{\sum_{j=1}^p \bar{y}^j \cdot \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]}{\sum_{j=1}^p \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]} \quad (5.46)$$

$$\text{Rede Neural } Y = \Gamma\left[W^3\Gamma\left[W^2\Gamma\left[W^1X\right]\right]\right] \quad (5.47)$$

onde as dimensões dos vetores e matrizes são:

$$Y(m), X(n), W^1(n,p), W^2(p,q) \text{ e } W^3(q,m)$$

Sendo que $\Gamma[\bullet]$ é uma função não linear, normalmente a função sigmóide.

Com base nas expressões acima, é possível fazer uma análise da complexidade dos algoritmos, relacionando: o número de operações básicas envolvidas; número de cálculos de funções e número de parâmetros. Esta metodologia de análise comparativa é mais significativa do que comparar tempo de processamento, pois o mesmo depende do software e hardware utilizados.

A tabela abaixo resume o número de operações envolvidas utilizando as três estruturas. A rede neural é definida pelos parâmetros $[n, p, q, m]$, sendo n o número de entradas, p o número de neurônios da primeira camada oculta, q o número de neurônios da segunda camada oculta, e m o número de saídas. Para a rede do Wang e o NFNG p significa o número de funções de pertinência.

Tabela 5.1 - Estudo comparativo entre as três estratégias de identificação.

Estrutura	Operações de Multiplicação	Operações de soma/ subtração	Operações de divisão	cálculos de funções	Número de Parâmetros
Rede Neural [n, p, q, m]	$n.p + p.q + q.m$	$n.p + p.q + q.m$	Nenhuma	$p+q+m$ (função sigmóide)	$n.p + p.q + q.m$ w_{ij}^1, w_{ki}^2 e w_{lk}^3
Rede Wang [n, p, m]	$(n.p + p).m$	$(n.p+p).m$	$(n.p+p).m$	$n.p.m$ (função gaussiana)	$2.n.p.m+p.m$ \bar{y}^j, \bar{x}_i^j e σ_i^j
NFNG [n, p, m]	$n.m$	$2.n.m$	nenhuma	$2.n$ (função linear)	$n.p.m+3n.p$ w_{ijl} e a_{ij}, α_{eij} e α_{dij}

Para a tabela acima, nas colunas operações não estão incluídas aquelas usadas para calcular as funções.

Com os dados mostrados na tabela 5.1 é possível fazer uma comparação entre as 3 estruturas na identificação dos sistemas dinâmicos descritos nos exemplos apresentados. Para exemplificar, considere o sistema descrito no exemplo 2. Para os resultados apresentados nas figuras 5.13, 5.14 e 5.15, a tabela 5.2 resume as operações efetuadas para as 3 estratégias.

Tabela 5.2 - Número de operações na identificação do sistema dinâmico do exemplo 2.

Estrutura	Operações de Multiplicação	Operações de somas/ subtração	Operações de divisão	cálculos de funções	Número de Parâmetros
Rede Neural [2,20,10,1]	250	250	nenhuma	31 (função sigmóide)	250 w_{ij}^1, w_{ki}^2 e w_{lk}^3
Rede Wang [2,40,1]	120	120	120	80 (função gaussiana)	120 \bar{y}^j, \bar{x}_i^j e σ_i^j
NFNG [2,5,1]	10	20	nenhuma	4 (função linear)	40 $w_{ijl}, a_{ij}, \alpha_{eij}$ e α_{dij}

Como pode ser verificado, o número de operações e cálculo de funções do NFNG é bem menor que o das outras duas estruturas. Além disto, as funções de pertinência calculadas envolvem uma operação de soma e uma de multiplicação para cada cálculo. Como as funções são complementares para os quatro cálculos são necessárias quatro operações de soma e duas de multiplicação. Levando em consideração estes resultados pode-se inferir que o tempo de processamento do NFNG é muito inferior, comparando com as outras duas configurações.

5.6 - CONCLUSÃO

Foi apresentada neste capítulo uma estrutura de neurônio *fuzzy* que mostrou-se bastante atraente em aplicação de identificação de sistemas dinâmicos. Foram obtidos resultados de identificação de sistemas dinâmicos semelhantes as redes neurais multicamadas e *neurofuzzy* do Wang (1994). A vantagem do NFNG sobre essas duas estruturas é que o número de operações envolvidas no seu processamento é consideravelmente menor e exige cálculos de funções mais simples. Isso faz com que o NFNG seja uma opção atraente em aplicações em tempo real e em treinamento *on-line*. Os resultados apresentados não permitem fazer uma generalização quanto à capacidade de aproximação e generalização usando o NFNG, mas servem para alertar que nem sempre é preciso utilizar estruturas complexas, como as redes neurais multicamadas e *neurofuzzy*, para identificar sistemas dinâmicos.

CAPÍTULO 6

DETECÇÃO E DIAGNÓSTICO DE FALHAS EM SISTEMAS DINÂMICOS

6.1 - INTRODUÇÃO

Neste capítulo, a detecção de falhas em sistemas dinâmicos é abordada. A idéia básica é tratar o problema de detecção de falhas como um problema de classificação de padrões. As ferramentas desenvolvidas nos capítulos 2, 4 e 5 são usadas para detectar e diagnosticar falhas nos sistemas dinâmicos modelados no capítulo 3. As estratégias propostas são baseadas somente em sistema de classificação de padrões, utilizando informações de entradas e saídas da planta, e em geração residual juntamente com sistema de classificação de padrões (estratégias híbridas). A utilização de sistema de classificação de padrões juntamente com geração residual é justificada pelo fato da lógica de determinação da falha a partir do resíduo não ser uma tarefa trivial (Frank, 1990). Com a utilização da rede *neurofuzzy and/or* proposta no capítulo 2, por exemplo, a lógica é determinada automaticamente, via treinamento. Os resíduos são gerados utilizando os observadores de estados em modos deslizantes, projetados no capítulo 4, e estimação dos parâmetros do modelo da planta, que são os pesos do *neo-fuzzy-neuron-generalizado*, capítulo 5. Os parâmetros são atualizados *on-line*, sendo que as variações são usadas na geração residual.

O objetivo do capítulo é fazer uma avaliação das estruturas. São apresentados resultados de simulações de falhas para os três sistemas dinâmicos modelados no capítulo 3. Para o Sistema de Tanques Interativos são também apresentados resultados experimentais de detecção *off-line*. Também é mostrada uma aplicação do sistema de detecção de falhas, baseado em observadores em modos deslizantes, em controle com reconfiguração, para o sistema de acionamento C.C..

6.2 - ESTRUTURA GERAL DO SISTEMA DE DETECÇÃO E ISOLAÇÃO DE FALHAS EM SISTEMAS DINÂMICOS

Como foi visto no capítulo 1, a estrutura geral do sistema de detecção e diagnóstico de falhas é mostrada na figura 6.1. Esta estrutura foi usada para detectar e diagnosticar falhas nos três sistemas modelados no capítulo 3, para três estratégias diferentes. A diferença básica de uma para outra é a forma de geração do padrão, que é o vetor de entrada do sistema de classificação. Na primeira, o padrão é formado somente utilizando as entradas e saídas da planta. Já a segunda e terceira, chamadas de estratégias híbridas, são utilizados também resíduos obtidos via observadores de estados e identificação de parâmetros, respectivamente.

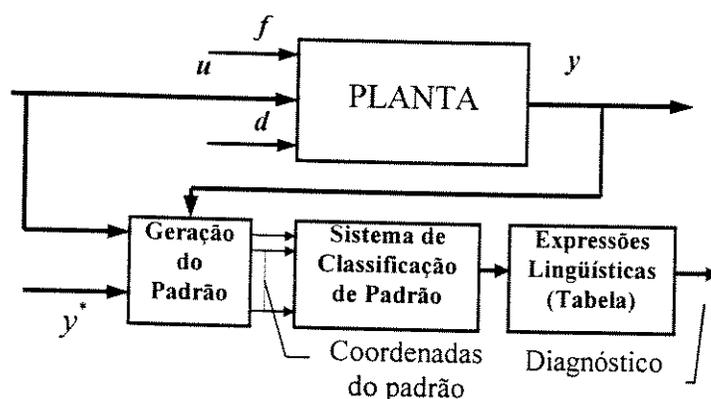


Fig. 6.1 - Estrutura do Sistema de Detecção e Identificação de Falhas

6.3 - ESTRATÉGIA BASEADA EM ENTRADAS E SAÍDAS DA PLANTA E SISTEMA DE CLASSIFICAÇÃO DE PADRÕES

Para esta estratégia o padrão é gerado a partir dos vetores u , y e y^* (para sistema em malha fechada). A principal vantagem é que há a necessidade do modelo matemático do sistema somente na fase de treinamento da rede, isto se resultados experimentais da planta não estão disponíveis.

6.3.1 - DETECÇÃO E ISOLAÇÃO DE FALHAS NO SISTEMA DE ACIONAMENTO EM CORRENTE ALTERNADA

A estrutura descrita anteriormente foi usada para detectar e isolar falhas no Sistema de Acionamentos em Corrente Alternada (SACA), detalhado no capítulo 3. Esse sistema, mostrado na figura 6.2, é composto por uma fonte de alimentação (rede elétrica), um motor de indução tipo rotor em gaiola e uma carga mecânica. No capítulo 3, foram modeladas as falhas relacionadas à fonte, a saber: curto-circuito entre fase e neutro e falta de alimentação, ou abertura de fase, representadas pelas chaves ideais. A estrutura do sistema de detecção e isolação de falhas é mostrado na figura 6.3.

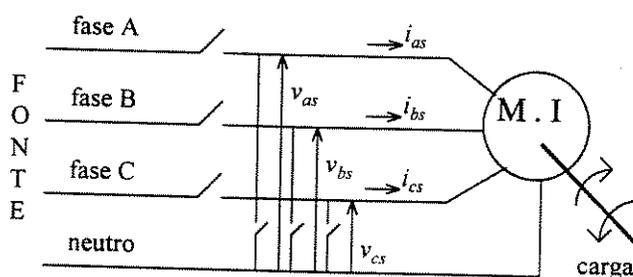


Fig. 6.2 - Sistema de acionamento C.A.

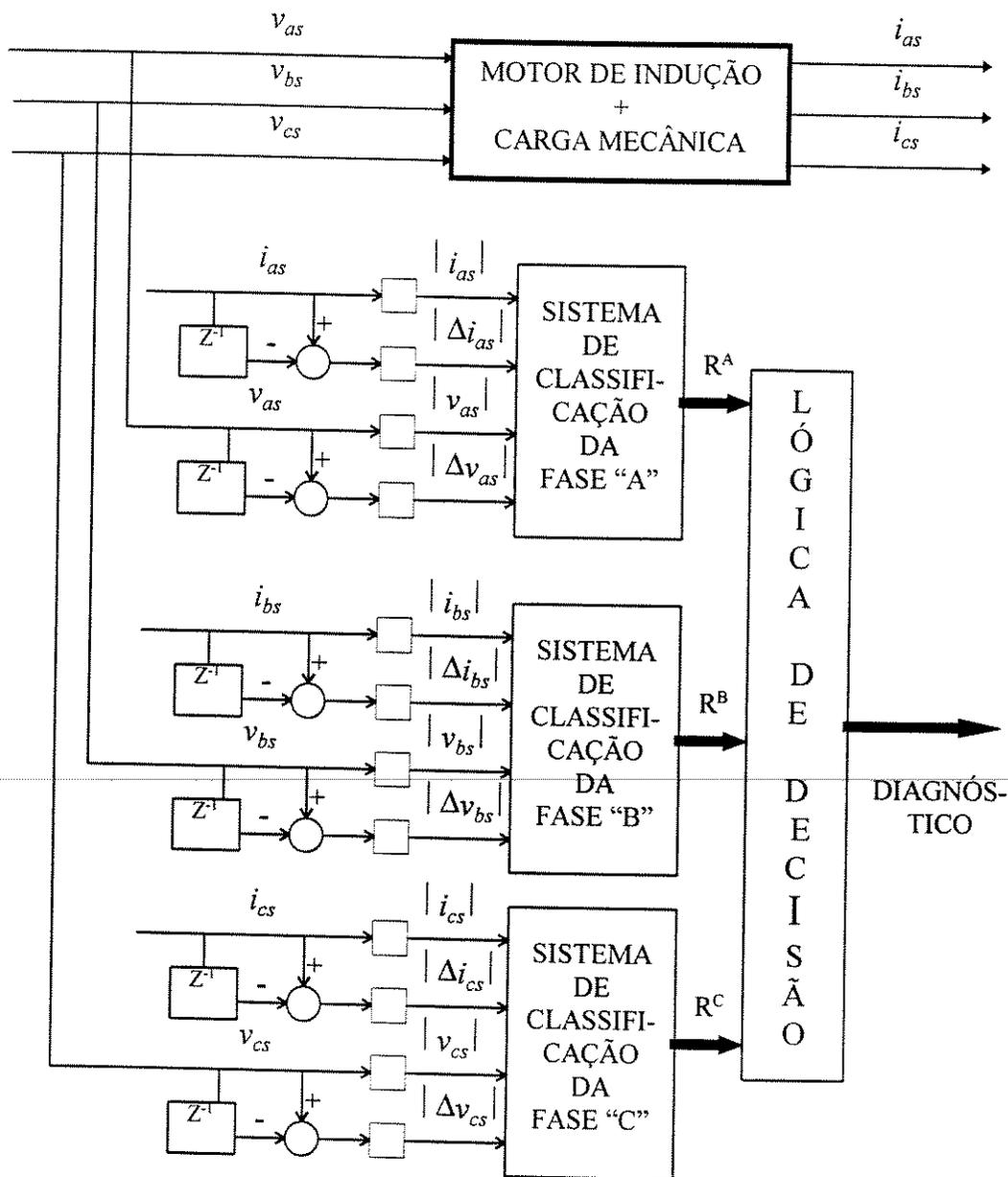


Fig. 6.3 - Estrutura do Sistema de Detecção e Isolação de Falhas no SACA

É utilizado um sistema de classificação para cada fase. Com os resultados dos três sistemas e a partir da lógica de decisão, tabela 1, tem-se o diagnóstico. As entradas dos sistemas de classificação, coordenadas dos padrões são: corrente; variação de corrente; tensão e variação de tensão de cada fase, respectivamente. Cada sistema de classificação possui três classes de saída: C_1 (fase operando normalmente), C_2 (curto-circuito da fase para o neutro) e C_3 (fase aberta/falta de alimentação). As saídas dos sistemas A, B e C correspondem aos índices das classes vencedoras, ou seja, 1, 2 ou 3.

Tabela 1. Lógica de Decisão

Índice da Falha	R ^A	R ^B	R ^C	Diagnóstico da Falha
01	1	1	1	operação normal
02	2	1	1	fase "A" em curto-circuito
03	1	2	1	fase "B" em curto-circuito
04	1	1	2	fase "C" em curto-circuito
05	2	2	1	fases "A,B" em curto-circuito
06	2	1	2	fase "A,C" em curto-circuito
07	1	2	2	fase "B,C" em curto-circuito
08	2	2	2	fase "A,B,C" em curto-circuito
09	3	1	1	fase "A" aberta
10	1	3	1	fase "B" aberta
11	1	1	3	fase "C" aberta
12	3	3	1	fases "A,B" abertas
13	3	1	3	fases "A,C" abertas
14	1	3	3	fases "B,C" abertas
15	3	3	3	fases "A,B,C" abertas

• Resultados Obtidos

Para avaliar o desempenho do sistema de detecção e diagnóstico de falhas da figura 6.3 foram feitas duas mil simulações de falhas, antes e depois do treinamento da rede. Para simular o sistema foi adotado o modelo do motor de indução para operação desequilibrada apresentado no capítulo 3, obedecendo à seguinte metodologia: tempo máximo de 1 segundo; o índice da falha (tabela 1) foi gerado de forma aleatória, para cada simulação; o tempo de ocorrência da falha gerado aleatoriamente; distúrbio de carga de amplitude aleatória, cujo valor máximo é 50% do nominal e ruído de medição aleatório e de amplitude 2% do valor nominal da grandeza. Na tabela 2 são mostrados resultados para sistemas de classificação baseados nas redes *neural* e *neurofuzzy*.

Tabela 2: Medida de desempenho do sistema de detecção e diagnóstico de falhas no acionamento C.A.

ÍNDICE DE DESEMPENHO	Rede <i>Neurofuzzy</i> : antes do treinamento	Rede <i>Neurofuzzy</i> : depois do treinamento	Rede Neural: depois do treinamento
alarme falso	0,2%	0,0%	0,0%
falha não detectada	0,5%	0,0%	0,0%
deteção detectada mas isolada incorretamente	0,3%	0,0%	0,0%
tempo médio de deteção da falha	≈4 ms	≈4 ms	≈4 ms

As redes neurais utilizadas para cada fase foram iniciadas com 50 neurônios por classe. A norma para determinação do neurônio vencedor foi a norma N_1 .

A topologia inicial da rede *neurofuzzy* usada, para cada fase, foi: quatro coordenadas de entrada; duas partições *fuzzy* para cada entrada (pequeno, P e grande, G); $2^4 = 16$ neurônios tipo *and*, sendo que três foram eliminados após o treinamento; três neurônios tipo *or*; s-norma=max e t-norma=produto; $\alpha_1=0.001$, $\alpha_3=0.1$ e $\alpha_2=\alpha_4=0$. O tempo de processamento do sistema de detecção e diagnóstico de falhas da figura 6.3, utilizando um microcomputador 586 - 133 MHz é inferior a $100\mu s$, que permite implementação em tempo real.

Comparando o treinamento das duas estruturas de redes, verificou-se que o tempo gasto para o aprendizado da rede neural foi de aproximadamente 30 vezes ao da rede *neurofuzzy and/or*, para obter os resultados da tabela 2. Este fato foi decisivo para escolher a rede *neurofuzzy* para os exemplos que serão mostrados daqui para frente.

Como pode ser verificado nos resultados da tabela 2, para a rede *neurofuzzy*, somente com a inicialização dos pesos o sistema consegue classificar corretamente as falhas, com um tempo médio de *detecção* de aproximadamente dois períodos de amostragens, que é de 2ms. O mesmo ocorre para a rede neural após o treinamento.

As figuras 6.4 a 6.9 ilustram o desempenho do sistema de detecção de falhas. A situação mostrada corresponde ao funcionamento normal do acionamento seguido de um curto-circuito das fases "A" e "B" para o neutro.

Os resultados mostrados na tabela 2 são justificados pelas características das falhas tratadas, que são bruscas e bem definidas no espaço formado pelas variáveis de entrada do sistema de classificação. Considerando a figura 1.2 do capítulo 1, para as falhas da tabela 1 os pontos de funcionamento do sistema durante as falhas estão situados sobre os eixos coordenados, por exemplo, um curto-circuito da fase "i" para o neutro leva os valores $|v_{is}|$ e $|\Delta v_{is}|$ próximos a zero, figuras 6.4 e 6.6. Nestes casos os pontos indicadores das falhas estão distantes da superfície de classificação, logo, mesmo que esta superfície não seja muito próxima da fronteira de separação real, a possibilidade de ocorrer alarmes falsos, detecção incorreta e não detecção da falha é reduzida. Já o tempo de *detecção* é pequeno porque as falhas são bruscas e a dinâmica do sistema é rápida. Este fato também pode ser visualizado a partir dos valores das saídas das redes *neurofuzzy* (Z_i), figura 6.7 a 6.9. O valor da saída z_i indica qual é o grau com que o padrão X_p pertence à classe C_i . A classe vencedora é aquela que possui o maior grau de pertinência. Quanto maior for a predominância do grau de pertinência da classe vencedora sobre as demais classes, mais confiável é a classificação. No caso de problema de detecção de falhas, a situação onde duas ou mais classes apresentam valores de graus de pertinência muito próximos, a possibilidade de alarme falso, não detecção ou detecção incorreta da falha é maior. A utilização da rede *neurofuzzy and/or* permite obter a informação dos graus de pertinência de todas as classes, com isso, ela fornece não só a falha mais provável (classe vencedora) como um conjunto de possíveis falhas no sistema. Sendo, portanto, uma informação mais "rica". Observando as figuras, verifica-se que antes da falha o valor da saída, z_1 , que corresponde à classe 1, operação normal, é praticamente 1, e para as demais classes os valores de z_i são praticamente nulos. Nota-se que, após dois períodos de amostragem da ocorrência da falha, para as redes das fases "A" e "C", o valor de z_1 é praticamente nulo, e o de z_2 , da classe 2 (indicadora de curto-circuito), torna-se praticamente 1, sendo que para as demais classes não há alteração. Este fato foi verificado para as demais falhas.

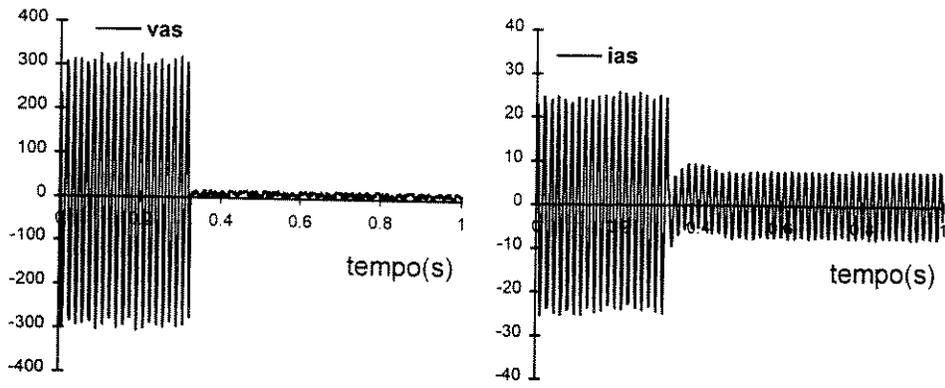


Fig. 6.4 - Tensão e corrente na fase "A" do motor de indução

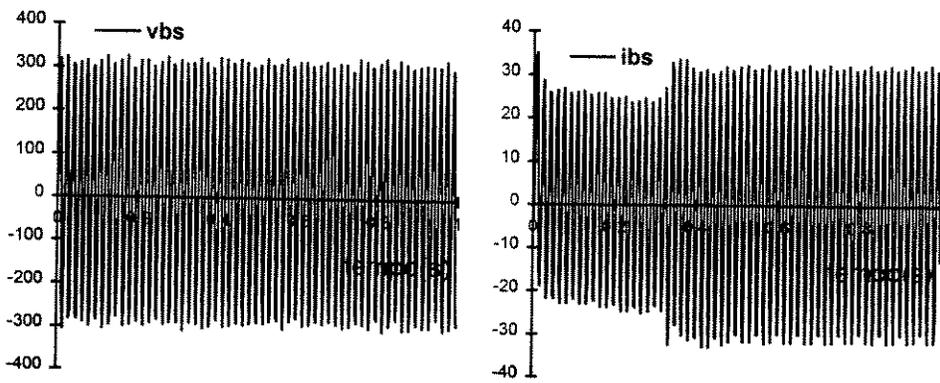


Fig. 6.5 - Tensão e Corrente na fase "B" do motor de indução

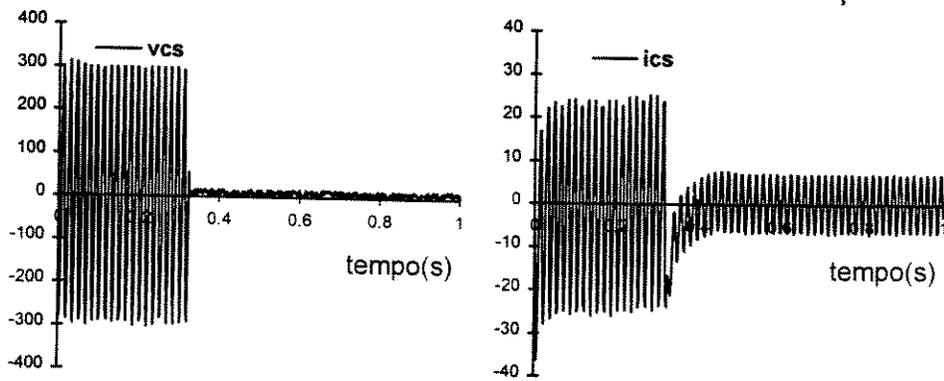


Fig. 6.6 - Tensão e Corrente na fase "C" do motor de indução

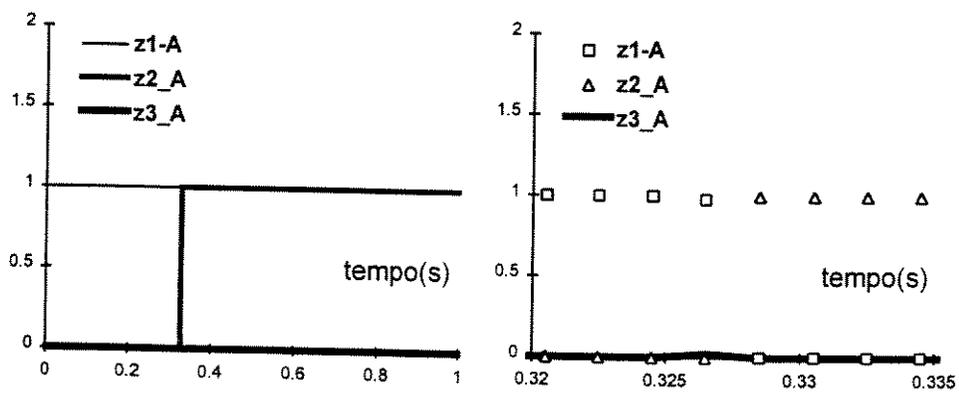


Fig. 6.7 - Saídas da rede neurofuzzy da fase "A"

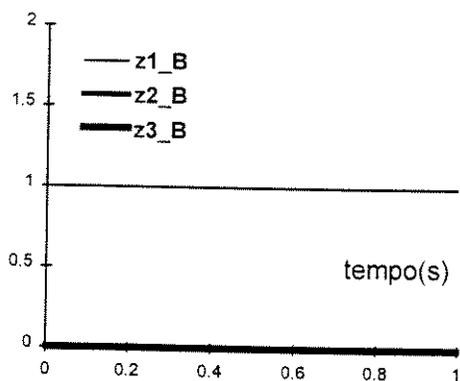


Fig. 6.8 - Saídas da rede *neurofuzzy* da fase “B”

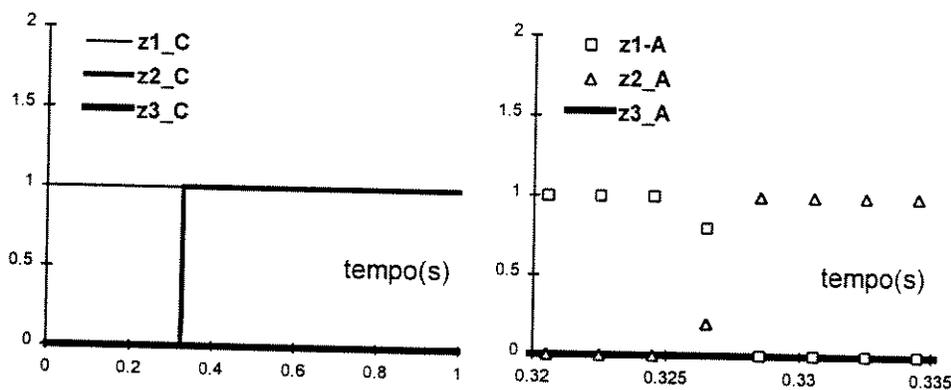


Fig. 6.9 - Saídas da rede *neurofuzzy* da “C”

• **Modelagem Qualitativa das Falhas**

Uma informação que pode ser extraída automaticamente da rede *neurofuzzy* é o modelo qualitativo, baseado em regras *fuzzy*, tabela 3. Na tabela, *P* significa pequeno e *G* grande, sendo *P* e *G* definidos por funções de pertinência como mostradas no capítulo 2. Após extrair as regras não utilizadas, tem-se o seguinte modelo qualitativo:

Tabela 3: Modelo qualitativo para falhas na fase A

se Δi_{as} é:	se Δv_{as} é:	se i_{as} é:	se v_{as} é:	Então a fase está :
<i>P</i>	<i>P</i>	<i>P</i>	<i>G</i>	<i>aberta</i>
<i>P</i>	<i>P</i>	<i>G</i>	<i>P</i>	<i>em curto</i>
<i>P</i>	<i>G</i>	<i>P</i>	<i>P</i>	<i>aberta</i>
<i>P</i>	<i>G</i>	<i>P</i>	<i>G</i>	<i>aberta</i>
<i>P</i>	<i>G</i>	<i>G</i>	<i>P</i>	<i>em curto</i>
<i>P</i>	<i>G</i>	<i>G</i>	<i>G</i>	<i>normal</i>
<i>G</i>	<i>P</i>	<i>P</i>	<i>P</i>	<i>em curto</i>
<i>G</i>	<i>P</i>	<i>G</i>	<i>P</i>	<i>em curto</i>
<i>G</i>	<i>P</i>	<i>G</i>	<i>G</i>	<i>normal</i>
<i>G</i>	<i>G</i>	<i>P</i>	<i>P</i>	<i>aberta</i>
<i>G</i>	<i>G</i>	<i>P</i>	<i>G</i>	<i>normal</i>
<i>G</i>	<i>G</i>	<i>G</i>	<i>P</i>	<i>normal</i>
<i>G</i>	<i>G</i>	<i>G</i>	<i>G</i>	<i>normal</i>

A utilização do sistema de acionamento C.A., além de permitir uma avaliação da estratégia de detecção, permite avaliar o modelo qualitativo. Um especialista montaria as seguintes regras para as falhas da tabela 1: se a corrente e variação de corrente são pequenas então a fase está aberta e se a tensão e variação de tensão são pequenas então a fase está em curto-circuito para o neutro. Nota-se que estas regras estão contidas na tabela 3, mostrando assim que o modelo qualitativo representa bem o conhecimento do especialista do processo.

6.3.2 - DETECÇÃO E ISOLAÇÃO DE FALHAS NO SISTEMA DE TANQUES INTERATIVOS

Para o Sistema de Tanques Interativos mostrado na figura 6.10, cujas falhas foram modeladas no capítulo 3, foram tratados três tipos de falhas: em atuadores; em componentes e em sensores.

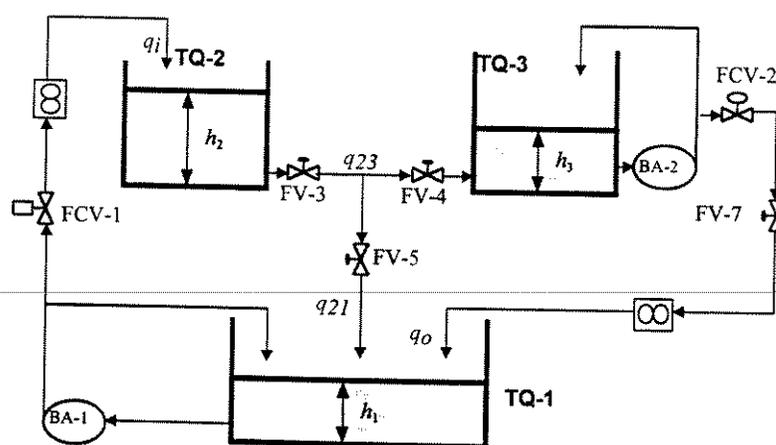


Fig. 6.10 - Sistema de tanques interativos

a) Sistema Completo para Detectar e Isolar Falhas em Atuadores, em Componentes da Planta e de Instrumentação

Inicialmente procurou-se utilizar uma estrutura de sistema de detecção de falhas com apenas um sistema de classificação. Foram utilizadas como coordenadas do padrão as saídas reais e de referência e os sinais de controle. Este sistema não proporcionou bons resultados, porque algumas falhas apresentavam comportamentos muito parecidos, conforme pode-se verificar nos resultados apresentados por Pereira (1996). Diante disso, partiu-se para a utilização do sistema de classificação de dois estágios, mostrado na figura 6.11.

Como foi dito no capítulo 1, se o número de *clusters* não é conhecido a priori, este pode ser tomado como sendo igual ao número de falhas do sistema, que correspondem aos neurônios da segunda camada. Durante o treinamento alguns neurônios podem ser eliminados. No final do treinamento o número de neurônios da segunda camada corresponde ao número de *clusters*.

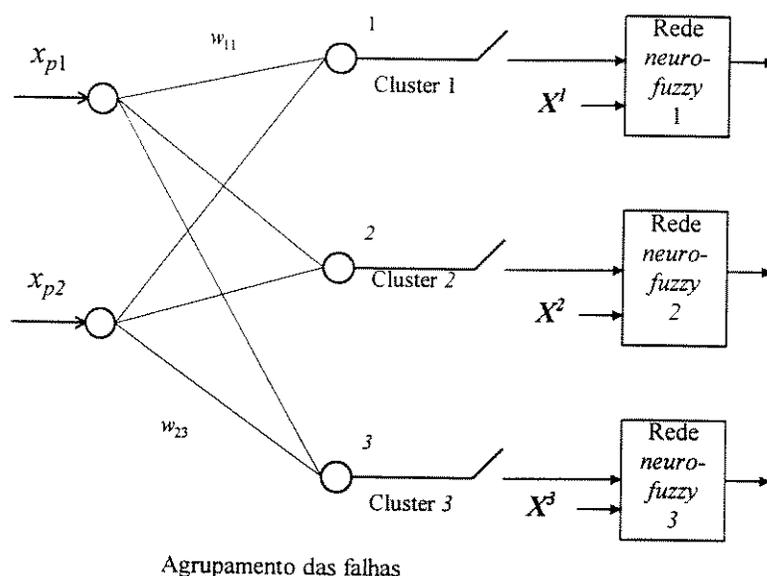


Fig. 6.11 - Sistema de detecção de falhas no STI: abordagem com 2 estágios

No caso do STI as falhas estão relacionadas às alterações de vazões de entrada do tanque TQ-2, saída de TQ-3 e entre TQ-2 e TQ3. Em Camargo (1997) são usadas variáveis auxiliares que são diferenças entre valores de vazões calculadas e medidas para fazer uma pré-seleção das falhas no sistema, agrupando-as em 3 classes. Aqui serão usadas as diferenças entre os sinais de controle e as vazões medidas, valores normalizados. Estas diferenças seriam iguais às variáveis auxiliares utilizadas por Camargo (1997) se o modelo da válvula fosse linear. Verificou-se via simulações que os *clusters* obtidos usando as duas abordagens foram os mesmos. Para entender como estas variáveis contêm informações de falhas no STI, considere a situação de agarramento da válvula CV-1. Quando CV-1 agarra numa posição que permite uma vazão maior do que a vazão requerida pelo controlador, o sinal de controle tenta diminuir a vazão, mas não o consegue devido ao agarramento da válvula, provocando uma variação na diferença entre o valor medido e calculado para a vazão de entrada de TQ-2.

A topologia da rede neural empregada no primeiro estágio foi:

Entrada:

$$X_p = [u_1 - y_3; \quad u_2 - y_2]$$

Saída: três neurônios, que correspondem ao *clusters*.

Utilizando o algoritmo 2, apresentado no capítulo 2, e empregando a norma N_1 para determinação do neurônio vencedor, chegou-se às seguintes configurações dos *clusters*, tabela 4.

Tabela 4: Configuração dos clusters após o treinamento da rede neural.

Cluster 1: Falhas	Cluster 2: Falhas	Cluster 3: Falhas
Entupimento do duto entre TQ-1 e TQ-2	Entupimento do duto entre TQ-2 e TQ-3	Entupimento do duto entre TQ-3 e TQ-1
Agarramento da válvula CV-1	Vazamento do duto entre TQ-2 e TQ-3	Agarramento da válvula CV-2
Desligamento da bomba BA-1	Falha no sensor de nível de TQ-3	Desligamento da bomba BA-2
Falha no sensor de vazão de entrada de TQ-2	Operação normal	Falha no sensor de vazão de saída de TQ-3

Após a determinação dos *clusters*, foram feitas mais duas mil simulações de falhas para verificar a qualidade do processo de clusterização. As faixas de variações dos parâmetros que indicam a intensidade da falha estão relacionadas no apêndice A. O resultado foi que 95,3% das falhas foram alocadas corretamente no *cluster* definido pela rede. O erro normalmente ocorreu para falhas que provocam pequenas variações no comportamento dinâmico do sistema, por exemplo, um entupimento do duto entre TQ-1 e TQ-2, que pertence ao *cluster* 1, de baixa intensidade foi confundido como operação normal que pertence ao *cluster* 2.

Um vez estabelecidos os *clusters*, os sistemas de classificação, segundo estágio, que são baseados em redes *neurofuzzy and/or* foram treinados separadamente. Como foi dito no capítulo 1, as redes *neurofuzzy* não necessariamente precisam ter as mesmas entradas. As topologias das redes empregadas foram:

$$\text{RNF1: Entradas } [u_1 - y_3 \quad y_3 \quad y_1 - y_1^*]$$

Saídas: os índices das falhas da primeira coluna da tabela 4

$$\text{RNF2: Entradas } [y_1 \quad y_1 - y_1^* \quad y_2 \quad y_3]$$

Saídas: os índices das falhas da segunda coluna da tabela 4

$$\text{RNF3: Entradas } [u_2 - y_2 \quad y_2 \quad y_2 - y_2^*]$$

Saídas: os índices das falhas da terceira coluna da tabela 4

Na tabela 5 são mostrados resultados de simulação de falhas entre aquelas contidas na primeira coluna da tabela 4. Foram feitas dezesseis mil simulações de falhas. Para estas simulações o índice da falha, o instante de ocorrência e variações de setpoints foram gerados aleatoriamente.

Tabela 5: Desempenho do sistema de detecção de falhas do STI para falhas do cluster 1

Falhas	Número de Ocorrência	Percentual de detecção correta	Tempo médio de detecção
Entupimento do duto entre TQ-1 e TQ-2	4058	59,92 %	38,2 s
Agarramento da válvula CV-1	3994	56,11%	28,1 s
Desligamento da bomba BA-1	3940	49,49%	69,0 s
Falha no sensor de vazão de entrada de TQ-2	4008	100%	2,8 s

Os resultados apresentados na tabela 5 retratam a dificuldade de detecção de falhas de baixa intensidade, que não provocam variações bruscas das grandezas utilizadas como entradas do sistema de classificação. Para a falha do sensor de vazão, que é brusca, a detecção ocorreu em 100 % dos casos. Isto sugeriu o desenvolvimento de um sistema simplificado para detectar falhas de instrumentação, abordado a seguir.

b) Sistema Simplificado para Detecção e Isolação de Falhas de Instrumentação

Neste item é apresentada uma estrutura de sistema de classificação com um estágio para detecção de falhas de instrumentação. A estrutura da rede *neurofuzzy* utilizada possui as seguintes características:

- cinco entradas: $u_1; u_2; y_1^* - y_1; y_2^* - y_2, u_1 - y_3$. Sendo que todas as grandezas são normalizadas;
- duas partições *fuzzy*: pequeno e grande;
- quatro saídas (classes) que correspondem a: operação normal (classe 1); falha no sensor de nível (classe 2); falha no sensor de vazão de saída de TQ-3 (classe 3) e falha no sensor de vazão de entrada de TQ-2 (classe 4).

• Resultados de Simulações

Foram feitas várias simulações utilizando o modelo de falha apresentado no capítulo 3. A tabela 6 mostra os resultados de vinte mil simulações de falhas nos três sensores utilizando a estratégia proposta, antes e depois do treinamento da rede *neurofuzzy*, obedecendo a mesma metodologia adotada no caso anterior. Para efeito de comparação, também são mostrados resultados obtidos por Aguirre (Aguirre *et alii*, 1996), para as mesmas condições, onde foi empregada a técnica de detecção e isolação de falhas utilizando observadores dedicados (E2). Comparando os resultados, nota-se que, para esta planta específica, os

resultados obtidos neste trabalho, de modo geral, apresentam um melhor índice de desempenho.

Tabela 6: Medida de desempenho do sistema de detecção e isolamento de falhas: E1 - estratégia proposta; E2 - baseado em observadores dedicados (Aguirre et alii, 1996).

ÍNDICE DE DESEMPENHO	E1 Antes do Treinamento	E1 Depois do Treinamento	E2
alarme falso	0,0%	0%	1,5%
falha não detectada	0,0%	0%	1,0%
falha detectada mas isolada incorretamente	0,0%	0%	0%
tempo médio de detecção da falha no sensor de nível	10,6 segundos	10,5 s	6,0 s
tempo médio de detecção da falha no sensor de vazão de saída de TQ-3	4,0 segundos	4,0 s	170,0 s
tempo médio de detecção da falha no sensor vazão de entrada de TQ-2	21,4 segundos	21,1 s	6,1 s

• Resultados Experimentais

Para validar a estrutura proposta foram utilizados os resultados experimentais obtidos por Pereira (1996). Um destes resultados é mostrado na figura 6.12, onde foi provocada uma falha no sensor de vazão de saída de TQ-3. A falha ocorreu em $t=500$ segundos.

Vale ressaltar que a validação experimental da técnica foi feita de forma *off-line*, a partir de resultados obtidos por Pereira (1996). Como o tempo de processamento da rede é de alguns μs e o menor período de amostragem da planta é de dois segundos, a implementação prática em tempo real é viável.

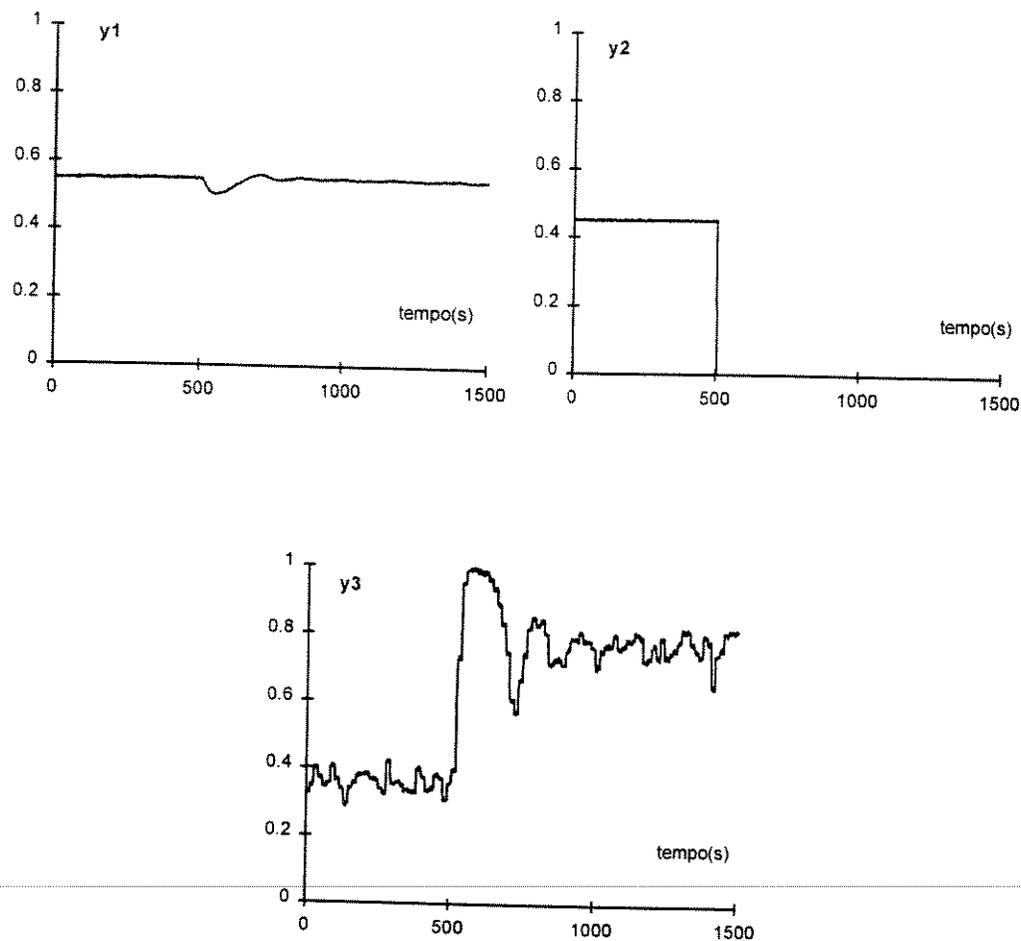


Fig. 6.12 - Resultado experimental para falha no sensor de vazão de saída de TQ-3 (Pereira, 1996).

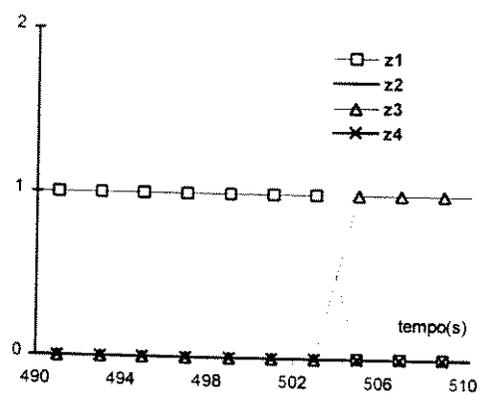


Fig. 6.13 - Saídas da rede *neurofuzzy and/or* para falha no sensor de vazão de saída de TQ-3.

• Modelo Qualitativo das Falhas de Instrumentação

O modelo qualitativo para falhas nos sensores de nível, vazão de saída de TQ-3 e vazão de entrada de TQ-2 é mostrado na tabela 7.

Tabela 7: Modelo qualitativo para falhas de instrumentação no Sistema de Tanques Interativos

se x_{p1} é	se x_{p2} é	se x_{p3} é	se x_{p4} é	se x_{p5} é	então a situação é
P	P	P	P	P	normal
P	P	P	P	G	falha no sensor 3
P	P	P	G	P	falha no sensor 2
P	P	G	P	P	normal
P	G	P	P	G	falha no sensor 3
P	G	P	G	P	falha no sensor 2
P	G	G	P	P	normal
G	P	P	P	P	normal
G	P	P	P	G	falha no sensor 3
G	P	G	P	P	falha no sensor 1
G	G	P	P	G	falha no sensor 3
G	G	P	G	P	falha no sensor 2
G	G	G	P	P	falha no sensor 1

6.4 - ESTRATÉGIA BASEADA EM OBSERVADORES DE ESTADOS EM MODOS DESLIZANTES E SISTEMA DE CLASSIFICAÇÃO DE PADRÕES

6.4.1 – FORMULAÇÃO DO PROBLEMA

Para facilitar o raciocínio, considere um sistema dinâmico linear sujeito a falha:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ed(k) + Jf(k) \\ y(k) &= Cx(k) + Fd(k) + Gf(k) \end{aligned} \quad (6.1)$$

onde $x \in \mathbf{R}^n$, $u \in \mathbf{R}^p$, $y \in \mathbf{R}^m$, $v \in \mathbf{R}^q$, $k \in \mathbf{N}$. Neste sistema, x representa o vetor de estados, u representa o vetor de entradas de controle (suposto conhecido) e y representa um vetor de medições disponível. As matrizes A , B e C são conhecidas e de dimensões apropriadas. O termo Jf representa as falhas nos atuadores e componentes da planta. Já Gf representa as falhas nos sensores de medição. Finalmente, os termos Ed e Fd representam as entradas desconhecidas, na planta e nos sensores de medição, respectivamente.

Como foi visto no capítulo 4, empregando o observador em modos deslizantes, calculado pela expressão:

$$\hat{x}(k+1) = \Phi_F \cdot \hat{x}(k) + \Phi_1 \cdot B \cdot u(k) + \Phi_2 \cdot y^1(k+1) - L \cdot y^r(k) \quad (6.2)$$

onde

$$\Phi_2 = H.(C^1.H)^{-L} \quad (6.3)$$

$$\Phi_1 = I - \Phi_2.C^1$$

$$\Phi_F = [\Phi_1.A + L.C^r] \quad (6.4)$$

Obtém-se um observador que é robusto a perturbações e/ou entradas desconhecidas nas direções do espaço de estado estabelecidas pela matriz H . Portanto, é possível obter observadores que sejam robustos a um conjunto de falhas (erro de estimação nulo) e sensíveis às outras (erro diferente de zero). Comparando valores observados por diferentes observadores é possível detectar as falhas no sistema, figura 6.14. Os resíduos r_i (diferença entre valor observado e medido) são usados como entradas do sistema de classificação.

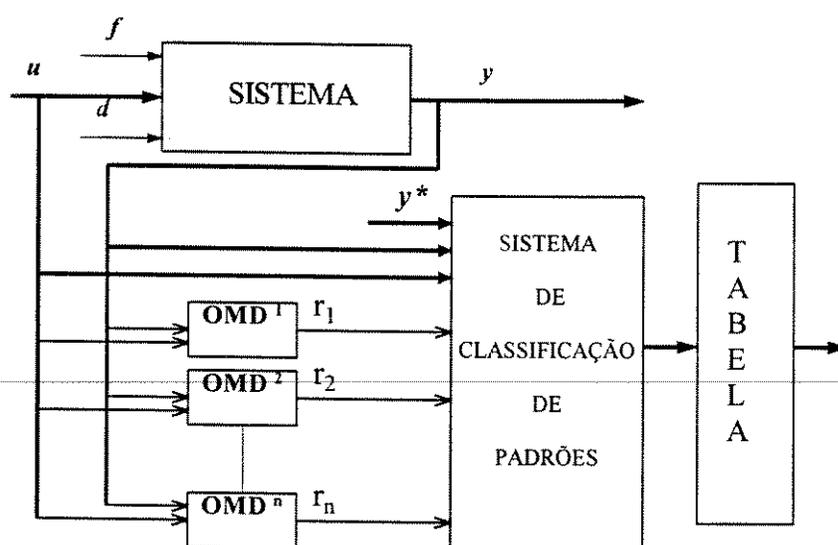


Fig. 6.14 - Geração de Padrão utilizando banco de observadores em modos deslizantes

6.4.2 – EXEMPLO DE PROJETO DE SISTEMA DE DETECÇÃO DE FALHAS UTILIZANDO OMD

Considere o sistema de acionamento C.C., figura 6.15, cujo modelo discreto é dado por:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} a_1 & a_2(k) & 0 \\ 0 & a_3 & 0 \\ a_4(k) & 0 & a_5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} [T_L(k)] \quad (6.5)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

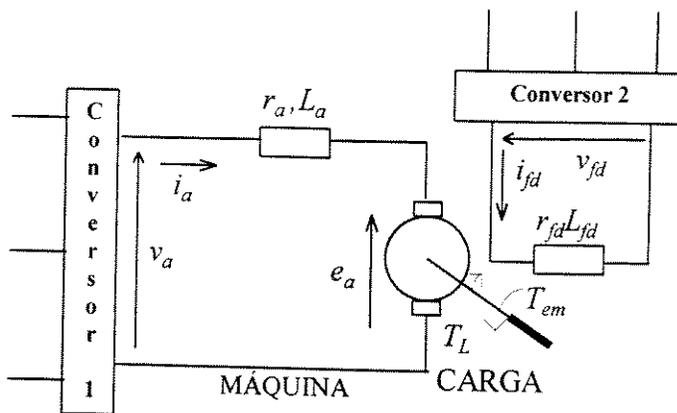


Fig. 6.15 - Sistema de acionamento C.C.

Como exemplo, será projetado um sistema capaz de detectar falhas nas direções das variáveis x_1 e x_2 e que seja robusto às variações de conjugado de carga, distúrbio mais freqüente. Para este fim, serão empregados os observadores projetados no capítulo 4. Uma falha na direção de x_1 , por exemplo curto-circuito da fonte de alimentação, não afetaria o observador que rejeita perturbações nas direções x_1 e x_3 (OMD¹³) e provocaria erros entre valores estimados e valores medidos para OMD²³, caso a saída y_2 não seja realimentada. Portanto, os erros (resíduos) dos dois observadores podem ser usados para detectar falhas no sistema. Considerando que para OMD¹³ e OMD²³ as saídas y_2 e y_1 não são realimentadas, respectivamente, tem-se as seguintes equações:

OMD¹³

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} L_{11}(k) & 0 & L_{13}(k) \\ L_{21}(k) & a_3 & L_{23}(k) \\ L_{31}(k) & 0 & L_{33}(k) \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fd}(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(k+1) \\ y_3(k+1) \end{bmatrix} - [L]y^r(k)$$

$$\begin{bmatrix} \hat{y}_1(k+1) \\ \hat{y}_2(k+1) \\ \hat{y}_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix}; \quad y^r(k) = [y_1(k) \quad y_3(k)]$$

(6.6)

OMD²³

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} a_1 & L_{12}(k) + a_2(k) & L_{13}(k) \\ 0 & L_{22}(k) & L_{23}(k) \\ 0 & L_{32}(k) & L_{33}(k) \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(k) \\ v_{fa}(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_2(k+1) \\ y_3(k+1) \end{bmatrix} - [L] [y^r(k)]$$

$$\begin{bmatrix} \hat{y}_1(k+1) \\ \hat{y}_2(k+1) \\ \hat{y}_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix}; \quad y^r(k) = [y_2(k) \quad y_3(k)]$$

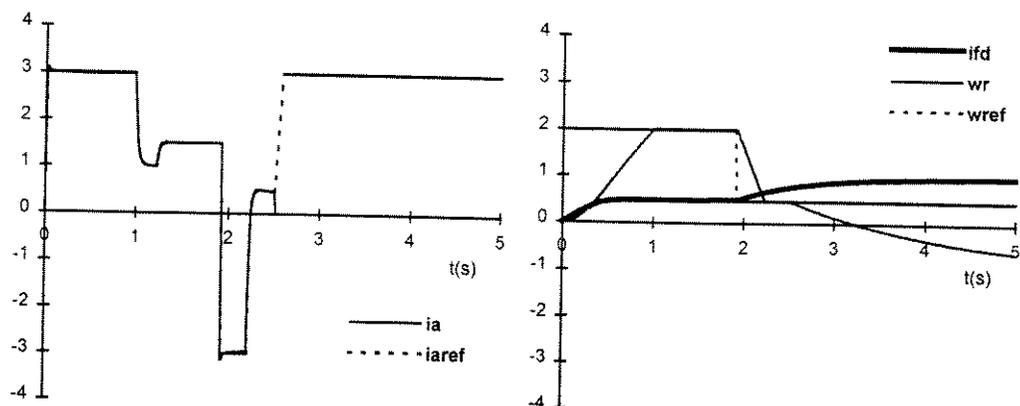
(6.9)

Com isto, definem-se os resíduos r_1 e r_2 que serão coordenadas do padrão, ou seja, entradas do sistema de classificação, da seguinte maneira:

$$\begin{aligned} r_1 &= \hat{y}_2^{OMD_{13}} - y_2 \\ r_2 &= \hat{y}_1^{OMD_{23}} - y_1 \end{aligned} \quad (6.10)$$

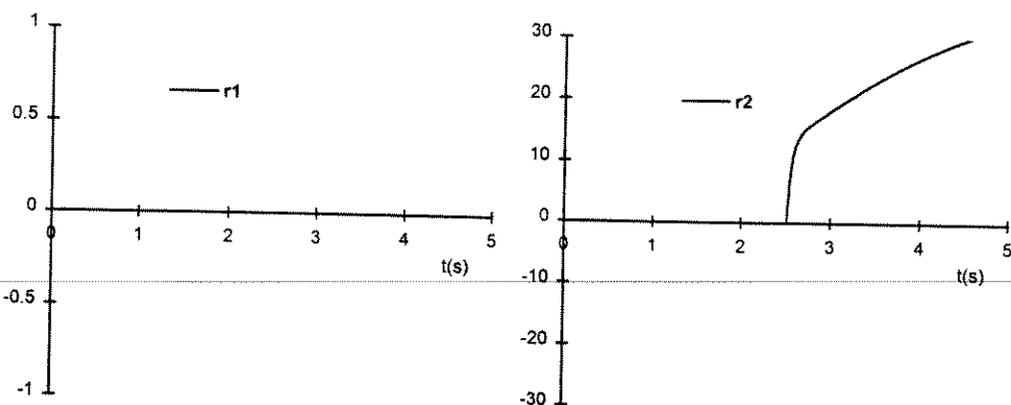
Para verificar o comportamento dinâmico dos resíduos, para ocorrência de falhas no sistema, são mostrados resultados de simulações de curto-circuito e desligamento das fontes de armadura e do campo. Os valores de velocidade de referência e conjugado de carga foram gerados de forma aleatória. O tempo de ocorrência da falha foi $t=2.5$ segundos, figura 6.16f. Os pólos do observador foram alocados em $[a1/2; a3; a5/2]$ para o observador OMD¹³ e em $[a1; a3/2; a5/2]$ para OMD²³. A escolha dos autovalores é importante, porque a dinâmica dos resíduos depende dos mesmos. A situação de funcionamento normal seguida de um desligamento da fonte da armadura é mostrada na figura 6.16a a 6.16f. Nota-se que, antes de ocorrer a falha, os resíduos r_1 e r_2 são nulos, mesmo quando ocorre variação do conjugado de carga, figura 6.16e, que é uma perturbação na direção de x_3 . Imediatamente após a ocorrência da falha, o resíduo r_2 , figura 6.16d, varia e r_1 permanece nulo, figura 6.16c, como era esperado, já que a falha é na direção de x_1 . Este fato se repete para o curto-circuito da armadura, que também é uma falha na direção de x_1 , figura 6.17. Quando ocorrem falhas na direção de x_2 , abertura e curto-circuito da fonte de campo, a situação se inverte, ou seja, r_1 varia e r_2 permanece em zero, conforme figuras 6.18 e 6.19.

No projeto do observador em modos deslizantes não foram consideradas perturbações nas variáveis medidas, portanto, quando estas ocorrem influenciam nos resíduos. Isto pode ser notado para uma falha no sensor de corrente de campo, $y_2(k)$, conforme figura 6.20. Neste caso, como a saída $y_2(k)$ não está sendo realimentada, o valor observado por OMD¹³ está correto. Por outro, o valor de y_2 medido é nulo, que é um valor incorreto. Isto provoca uma variação no resíduo r_1 . Este resultado sugere utilizar a estratégia de detecção de falhas, baseada em observador em modos deslizantes, para reconfigurar o sistema de controle na ocorrência de falhas nos sensores. Quando ocorrer um falha num determinado sensor, cuja variável associada é controlada, esta falha, sendo identificada, permitiria substituir o valor medido pelo valor observado na malha de controle. Este assunto será tratado mais a frente.



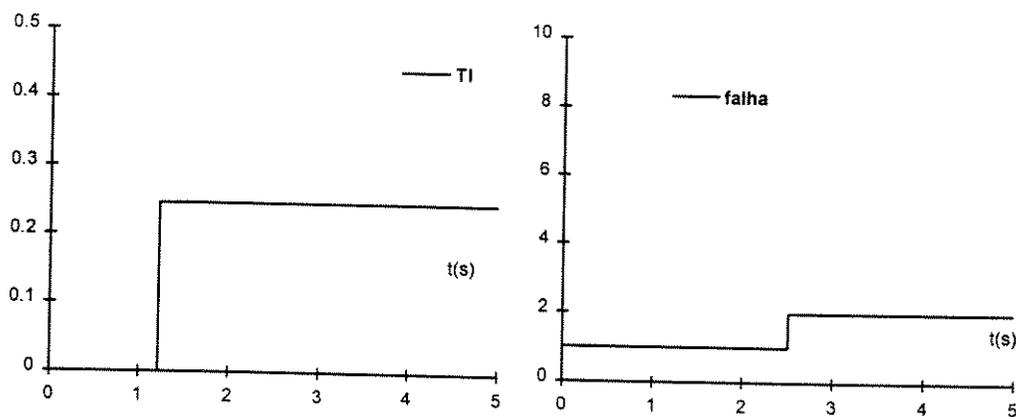
(a)

(b)



(c)

(d)



(e)

(f)

Fig. 6.16 – Simulação de falha no sistema de acionamento C.C.: abertura da fonte da armadura

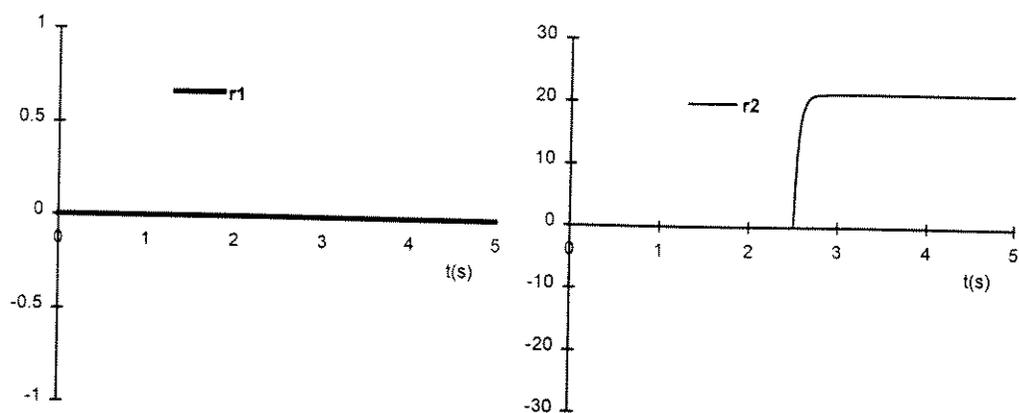


Fig. 6.17 – Simulação de falha no sistema de acionamento C.C.:
Curto-circuito da fonte de armadura

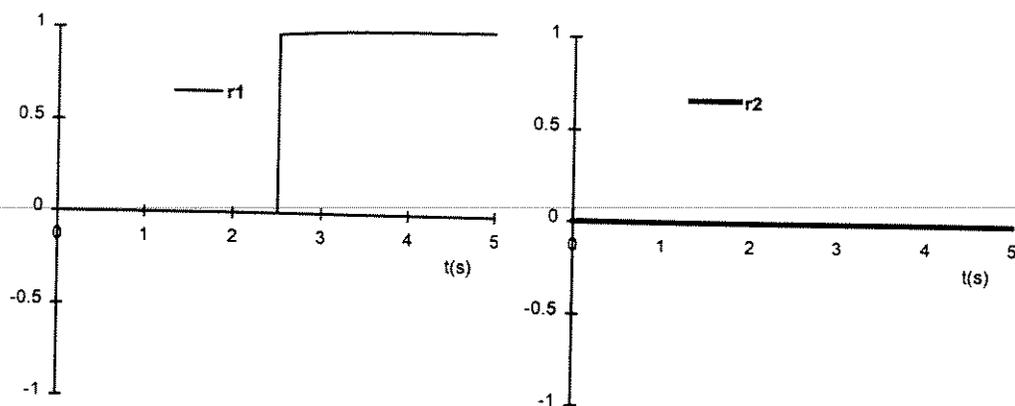


Fig. 6.18 – Simulação de falha no sistema de acionamento C.C.:
Curto-circuito da fonte de campo

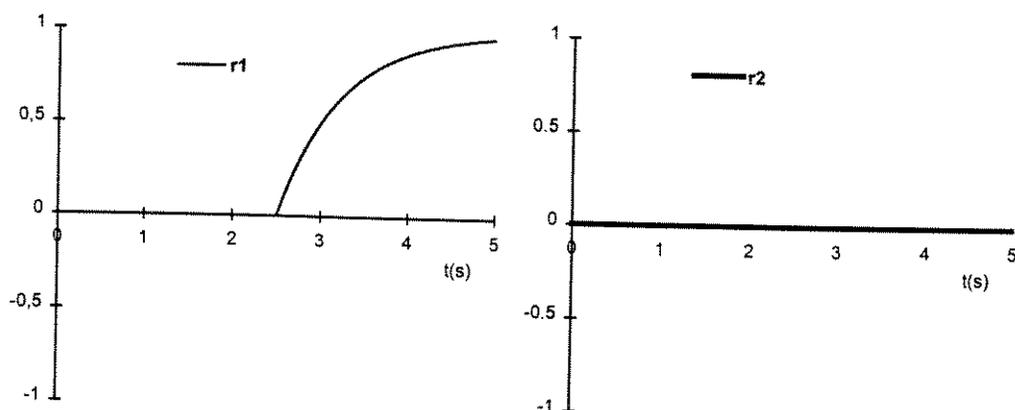


Fig. 6.19 – Simulação de falha no sistema de acionamento C.C.:
abertura da fonte de campo

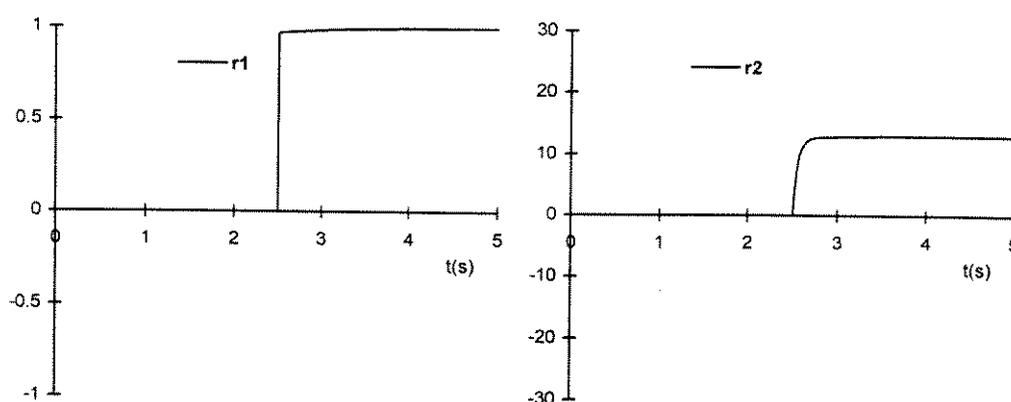


Fig. 6.20 – Falha no sistema de acionamento C.C.:
Falha do sensor de corrente de campo

Esses resultados mostram que os resíduos podem ser utilizados para detectar as falhas de forma eficiente. Por outro lado, verifica-se que somente com os valores dos mesmos não é possível isolar (classificar) as falhas. Comparando as figuras nota-se que o comportamento dos resíduos para uma abertura ou desligamento é praticamente o mesmo. Com a utilização dos valores das correntes de armadura e campo e velocidade, juntamente com os resíduos, é possível efetuar a classificação das falhas. Foi utilizado um sistema de classificação *fuzzy*, implementado pela rede *neurofuzzy and/or*, que possui a seguinte topologia:

Entradas: $[r_1(k) \quad r_2(k) \quad y_1(k) \quad y_2(k) \quad y_3(k)]$

Número de partições *fuzzy* para cada entrada: 3

Número de saídas: 8 (classes, índice da falha, tabela 8)

Tabela 8: Falhas nos sistema de acionamento C.C. usadas para testar o desempenho do sistema de detecção e diagnóstico de falhas baseado em observadores em modos deslizantes.

Índice da Falhas (Saída do sistema de classificação <i>fuzzy</i>)	Descrição das falhas
1	Operação normal
2	Abertura da fonte da armadura
3	Curto-circuito da fonte da armadura
4	Abertura da fonte do campo
5	Curto-circuito da fonte do campo
6	Falha no sensor da corrente da armadura
7	Falha no sensor da corrente do campo
8	Falha no sensor de velocidade

Para testar o desempenho do sistema de detecção de falhas foram feitas 2.000 simulações de falhas, cujo índice foi gerado aleatoriamente. O procedimento adotado foi o mesmo das simulações dos sistemas de detecção do STI e SACA: distúrbio de carga e referência de velocidade gerados aleatoriamente, tempo de ocorrência e valor. O resultado é mostrado na tabela 9.

Tabela 9: Resultado da detecção de falhas no sistema de acionamento C.C. usando observador em modos deslizantes.

	ÍNDICE DE DESEMPENHO
alarme falso	0,0%
Falha não detectada	0,7%
Falha detectada mas isolada incorretamente	1,2%

6.4.3 - APLICAÇÃO DA ESTRATÉGIA DE DETECÇÃO DE FALHAS EM CONTROLE RECONFIGURADO DO SISTEMA DE ACIONAMENTO C.C.

Neste item será vista uma aplicação da estratégia de detecção de falhas em controle reconfigurado, figura 6.21. A idéia básica é substituir o valor medido pelo sensor faltoso pelo valor observado, robusto à tal falha, ou seja a saída y' , usada no controlador vai ser o valor medido ou estimado.

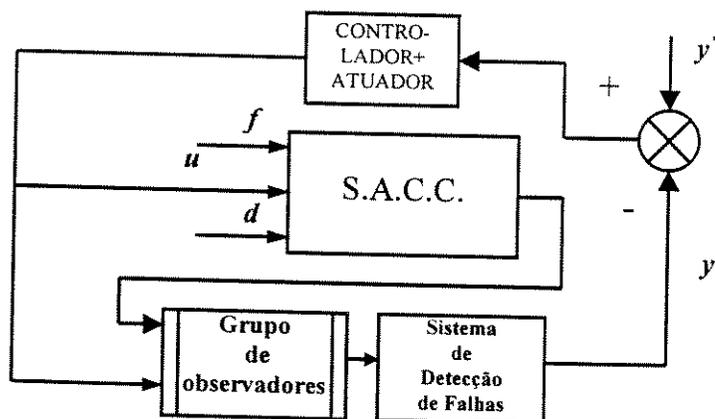


Fig. 6.21 - Estrutura do sistema de acionamento C.C. com controle reconfigurado.

Como mostrado no diagrama de blocos da figura 3.2, capítulo 3, para o sistema de acionamento C.C., as correntes da armadura e do campo, bem como a velocidade são variáveis controladas. Portanto, falhas nos sensores destas grandezas podem provocar erros grandes entre o valor de referência e o valor real. Para ilustrar, na figura 6.22a são mostradas as curvas de velocidade real e de referência para a situação de operação normal,

com a máquina funcionando numa velocidade que é o dobro da nominal, seguida de falha no sensor que mede a corrente de campo, em $t=2.5$ segundos. Como o valor da corrente de campo é nulo, que é menor que a referência, o controle atua no sentido de diminuir o erro, ou seja, o conversor de campo aumenta a tensão de campo. Isto acarreta um aumento da corrente de campo real, figura 6.21b, conseqüentemente, aumento do fluxo de campo. Este aumento do fluxo provoca uma redução da velocidade da máquina, que estabilizará num valor diferente da referência, com erro grande. Como foi visto no item anterior, quando ocorre uma falha no sensor de corrente de campo, o observador OMD¹³ não é afetado, porque esta grandeza não está sendo realimentada pelo mesmo. Neste caso, o valor observado é igual ao real, que por sua vez é diferente do valor medido, que está errado, figura 6.22a. A idéia da reconfiguração, já mencionada no item anterior, é que, uma vez detectada a falha no sensor, o valor medido é substituído pelo valor observado, que é o valor correto. Na figura 6.23 é mostrada a mesma simulação, utilizando a reconfiguração. Neste, a falha no sensor da corrente de campo foi detectada depois de 0.1 segundos. Observa-se que, depois que o valor observado é usado na malha de controle, o sistema volta à situação de erro nulo de velocidade. Devido a constante de tempo mecânica a velocidade praticamente não sente o efeito da falha durante os 0,1 segundos.

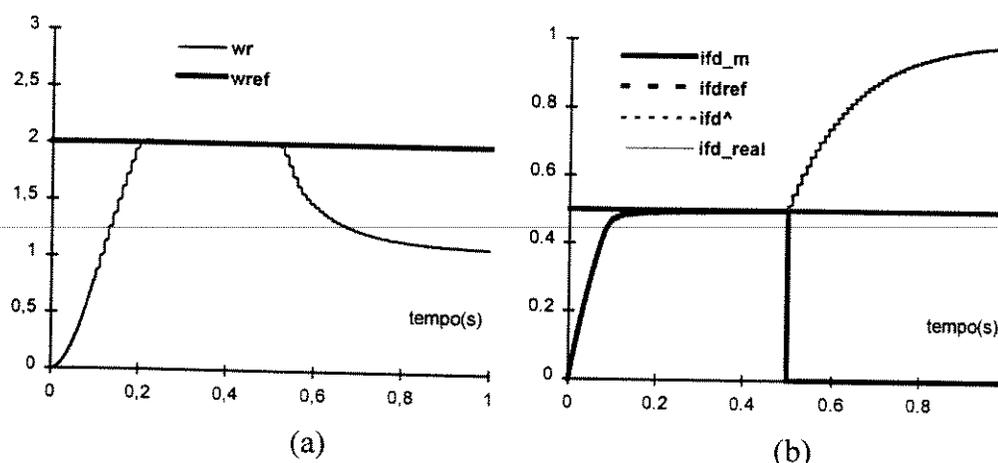


Fig. 6.22 – Simulação de uma falha no sensor de corrente de campo para o sistema sem reconfiguração

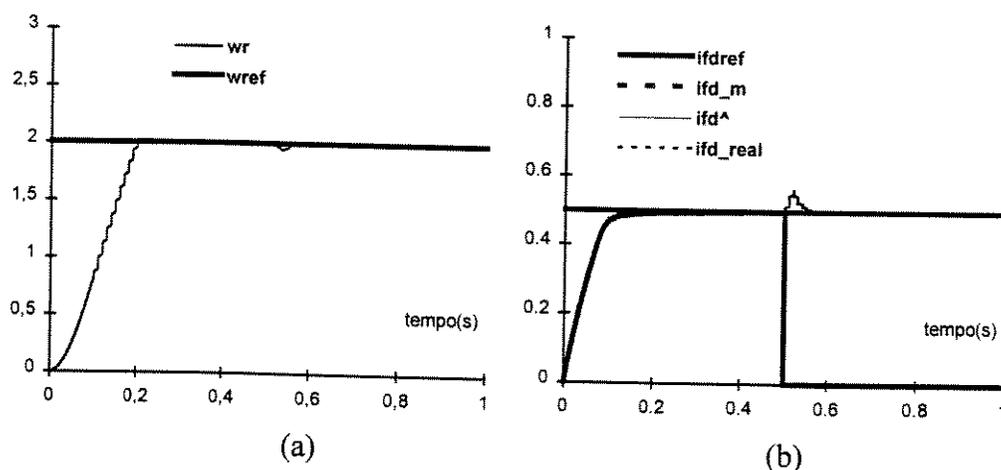


Fig. 6.23 – Simulação de uma falha no sensor de corrente de campo para o sistema com reconfiguração

6.5 - ESTRATÉGIA BASEADA EM ESTIMAÇÃO DE PARÂMETROS E SISTEMA DE CLASSIFICAÇÃO DE PADRÕES

No capítulo 5 foi demonstrado que, empregando um *neo-fuzzy-neuron-generalizado*, é possível encontrar um conjunto de pesos que permite efetuar um mapeamento arbitrário do espaço de entrada para o espaço de saída, com erro de aproximação nulo. Para exemplificar, considere-se um NFNG para identificação do sistema de acionamento C.C., que possui a seguinte estrutura:

$$\text{Entradas: } \begin{bmatrix} \hat{i}_a(k-1) & \hat{i}_{fd}(k-1) & \hat{w}_{k-1}(k-1) & v_a(k) & v_{fd}(k) \\ \hat{y}_1(k-1) & \hat{y}_2(k-1) & \hat{y}_3(k-1) & u_1(k) & u_2(k) \end{bmatrix} =$$

$$\text{Saídas: } [\hat{y}_1(k) \quad \hat{y}_2(k) \quad \hat{y}_3(k)]$$

Nas figuras 6.24 e 6.25 são mostradas as curvas da variável de saída y_1 (corrente de armadura) estimada e real, bem como o peso w_{121} do NFNG, para o primeiro e décimo ciclo de treinamento, respectivamente. Nota-se que no primeiro ciclo, a rede consegue aproximar com erro nulo, com grandes variações nos pesos. À medida que a rede aprende, estas variações dos pesos são menores. Para os resultados apresentados, o número máximo de partições *fuzzy* para cada entrada foi igual a três e as funções de pertinência são do tipo 2, capítulo 5. As entradas do sistema de acionamento C.C., u_1 e u_2 , foram geradas aleatoriamente.

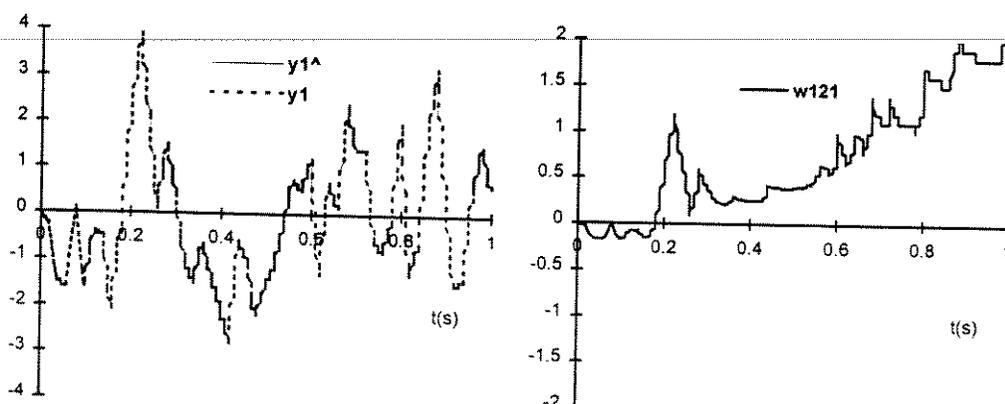


Fig. 6.24 – Saída y_1 estimada e real e peso w_{121} do NFNG: primeiro ciclo de treinamento

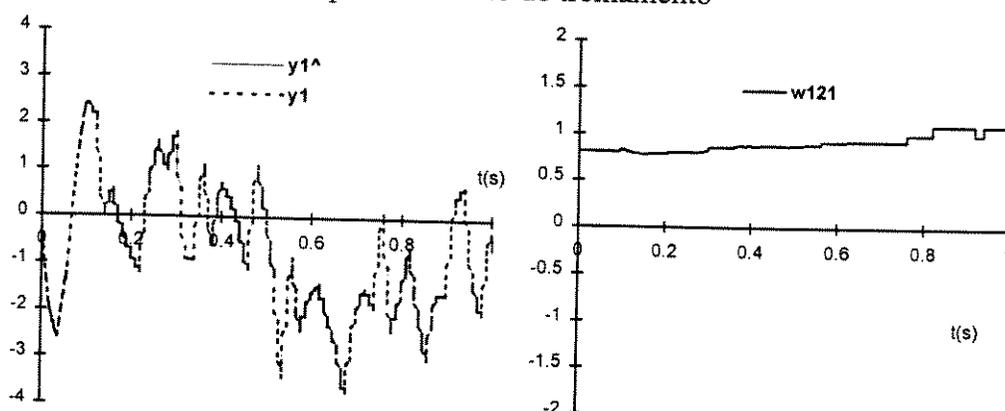


Fig. 6.25 – Saída y_1 estimada e real e peso w_{121} do NFNG: Décimo ciclo de treinamento

A topologia de detecção de falhas que utiliza a variação dos pesos do NFNG para geração de resíduos é mostrada na figura 6.26. Os pesos do NFNG W^0 (vetor) são determinados através do treinamento *off-line* da rede. Nesta etapa, os dados fornecidos para o treinamento são referentes às operações do sistema sem ocorrência de falhas. Já os pesos W são obtidos via treinamento *on-line*. Quando ocorrem falhas no sistema, os pesos W serão diferentes de W^0 , sendo que essa diferença é usada para gerar os resíduos, que são coordenadas dos padrões do sistema de classificação. Estas variações nos pesos são justificadas pelo fato que a falha provoca uma alteração do modelo que o NFNG tinha aprendido.

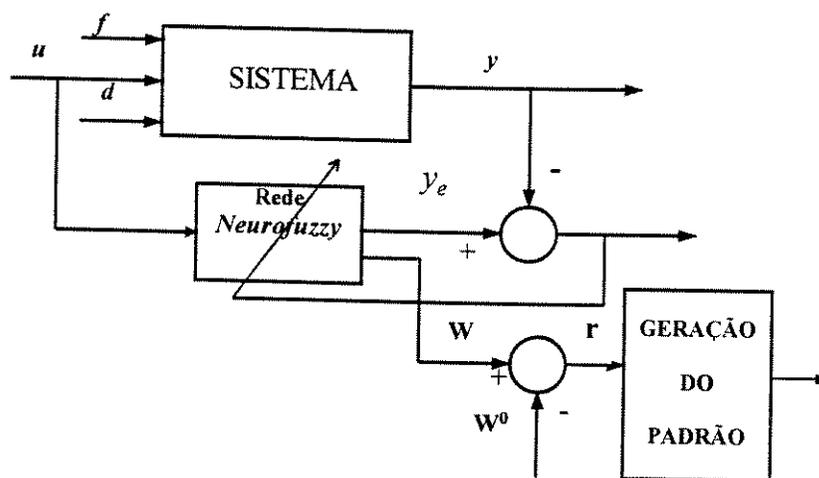


Fig. 6.26 - Geração de Padrão utilizando identificação de parâmetros

Para ilustrar, considere os seguintes resíduos:

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} w_{121} - w_{121}^0 \\ w_{222} - w_{222}^0 \\ w_{313} - w_{313}^0 \end{bmatrix} \quad (6.11)$$

lembrando que para o peso w_{ijl} , os subscritos i, j e l são os indexadores da entrada da rede, partição *fuzzy* e saída da rede, respectivamente. Os gráficos das simulações são mostrados nas figuras 6.27 a 6.30. Observa-se que, quando ocorre uma falha, em $t=0,5$ segundos, para as três simulações há variações nos pesos do NFNG. Na figura 6.29 são mostrados os resíduos para uma falha do sensor de velocidade. Como foi visto no capítulo 3, esta falha leva o valor da velocidade medida a zero. Quando ocorre a falha, há variações nos pesos do NFNG, principalmente naqueles que estão associados a saída y_3 , que é a velocidade. A abertura da fonte da armadura provoca variação brusca no peso que está relacionado a entrada $y_3(k-1)$ e a saída $y_3(k)$, figura 6.28. Para um curto-circuito da fonte de campo o peso mais afetado está relacionado a corrente de campo y_2 , figura 6.29. Na figura 6.30 é mostrada a curva do conjugado de carga para a situação de curto-circuito da fonte de campo. Nota-se que os resíduos são robustos às variações de carga. Para acontecer isto, no treinamento do NFNG este conjugado foi alterado de forma aleatória.

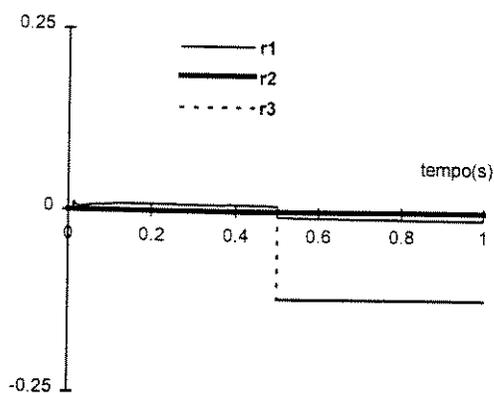


Fig. 6.27 – Comportamento dos resíduos para a falha no sensor de velocidade

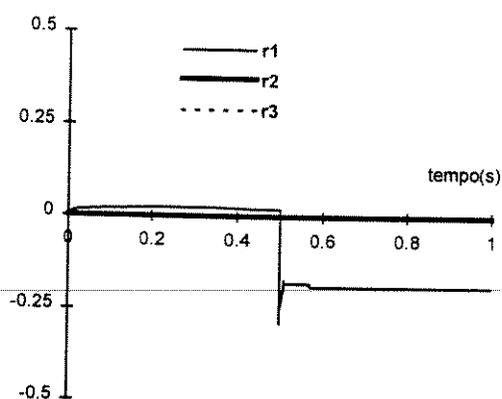


Fig. 6.28 – Comportamento dos resíduos para a falha: abertura da fonte da armadura

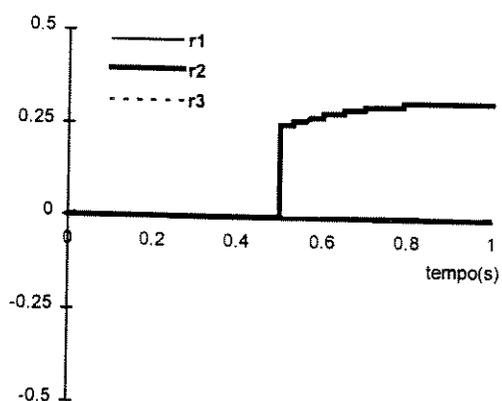


Fig. 6.29 – Comportamento dos resíduos para a falha: curto-circuito da fonte de campo

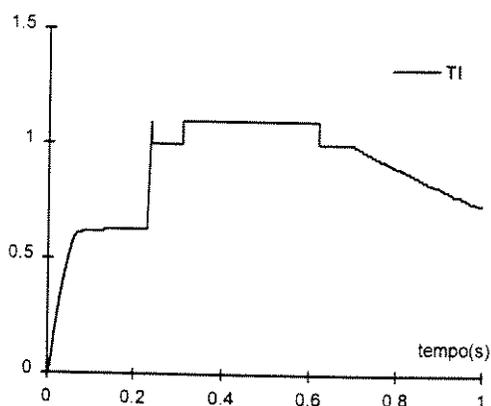


Fig. 6.30 – Curva do conjugado de carga para a falha: curto-circuito da fonte de campo

Da mesma forma que a estratégia anterior, só com as informações dos resíduos é possível apenas detectar a falha. Para permitir a isolação são necessárias outras informações, tais como valores das correntes e velocidade. Com isto, o sistema de classificação utilizado possui a seguinte topologia:

Entradas: $[r_1 \ r_2 \ r_3 \ x_1 \ x_2 \ x_3]$

Número de partições *fuzzy* para cada entrada: três

Número de saídas: oito (classes)

Os resultados das simulações mostram que para as falhas bruscas simuladas, as variações dos resíduos também são bruscas. Comparados com os resíduos gerados pelos observadores em modos deslizantes, nota-se que as variações dos mesmos são mais rápidas. Isto permite que as falhas sejam detectadas em menor espaço de tempo. Para efeito de comparação, uma falha no sensor de corrente de campo, que foi detectada em 100 ms utilizando a estratégia anterior, foi detectada em 8 ms utilizando a abordagem desse item.

6.6 - CONCLUSÃO

Foi proposta neste capítulo uma estrutura geral de um sistema para detecção e diagnóstico de falhas em sistemas dinâmicos, baseada em sistema de classificação em padrões. Foram vistos três formas de geração dos padrões: utilizando apenas informação de entrada e saída; através de resíduos obtidos via observadores de estados em modos deslizantes e através de identificação de parâmetros do modelo, empregando o NFNG. Cada estrutura possui vantagens e desvantagens. A primeira estrutura apresenta a vantagem de não necessitar do modelo matemático da planta, após o treinamento da rede. Por outro lado, a detecção de falhas que provocam variações lentas nas grandezas envolvidas ainda é um problema. Já a

estrutura baseada em observador de estados em modos deslizantes é interessante em aplicações como controle reconfigurado, embora necessite do modelo matemático. Já para a terceira estratégia, o modelo é obtido via treinamento do NFNG. Os coeficientes do modelo são os pesos do NFNG. As variações desses coeficientes são usadas na geração de resíduos, que são empregados para detectar as falhas no sistema. Por outro lado, sua aplicação em controle reconfigurado não é direta. Pode-se pensar em estrutura híbrida, que utilizaria o que tem de melhor em cada uma.

CAPÍTULO 7

CONCLUSÕES E CONSIDERAÇÕES FINAIS

A proposta do trabalho foi o desenvolvimento de estratégias de detecção e diagnóstico de falhas em sistema dinâmicos, sob o ponto de vista de classificação de padrões. Quatro tópicos diferentes foram explorados na abordagem do problema:

- a) classificação de padrões
- b) modelagem de falha
- c) observador de estado
- d) identificação de sistemas dinâmicos

O problema de *classificação de padrões* foi abordado sob a ótica de mecanismo de aprendizado. Foi proposta uma nova estrutura de rede *neurofuzzy*, bem como seu algoritmo de treinamento e processamento, batizada de rede *neurofuzzy and/or*, nome este que está relacionado com o tipo de neurônio empregado. As *vantagens* dessa rede são:

1. geração automática da topologia, com extração de conhecimento diretamente da base de dados; capacidade de aprendizado;
2. treinamento competitivo, sem uso de derivadas;
3. flexibilidade quanto à utilização de diversas **s** e **t**-normas;
4. tempo de processamento constante e tempo de treinamento reduzido com aumento da dimensão da rede, fixando-se o número de coordenadas de entrada;
5. possibilidade de extração de regras diretamente da topologia da rede.

No desenvolvimento do algoritmo de treinamento da rede, procurou-se tornar o processo de montagem do sistema de classificação o mais automático possível, desde a geração das funções de pertinência, que normalmente é uma tarefa difícil, até a obtenção da topologia final. Com isto, para efetuar a classificação de padrões, utilizando-se a rede *neurofuzzy and/or*, é necessário apenas um banco de dados contendo as coordenadas dos padrões, bem como sua classificação. A principal *desvantagem* da rede *neurofuzzy and/or* é que o algoritmo de processamento é de complexidade exponencial com relação ao número de entradas da rede.

Uma outra estrutura de rede abordada para *classificação de padrões* foi a rede neural auto-organizada. Para essa estrutura foi proposta uma metodologia de eliminação dos neurônios da rede em excesso. Essa metodologia mostrou-se eficiente, no que diz respeito à otimização da topologia da rede, como foi mostrado nos exemplos apresentados no capítulo 2. A principal vantagem da rede neural com relação à rede *neurofuzzy and/or* é que seu algoritmo de processamento é de complexidade polinomial. Por outro lado, se o número de entradas é fixado, o aumento do número de neurônios por classe implica em aumento de tempo de processamento, o que não ocorre para a rede *neurofuzzy and/or*. Além disso, a rede

neurofuzzy and/or permite não só uma classificação *crisp*, como é o caso da rede neural, mas também a informação do grau de pertinência do padrão para todas as classes de saída. Este resultado pode ser útil quando o sistema de classificação for usado em tarefa de diagnóstico, por exemplo. Os resultados apresentados mostraram que as estruturas apresentadas são alternativas viáveis para classificação de padrões.

Foram desenvolvidos *modelos matemáticos* que permitem simular *falhas* em três sistemas dinâmicos não lineares. O primeiro modelo de falhas desenvolvido foi de um sistema de acionamento elétrico empregando uma máquina de corrente contínua. Foram modeladas falhas nos conversores, que alimentam a armadura e o campo da máquina, e falhas na máquina propriamente dita. O segundo modelo foi de um sistema de acionamento empregando máquina de corrente alternada, especificamente a máquina de indução. Desenvolveu-se um modelo geral de falhas relacionadas à alimentação da máquina, que são curto-circuito e abertura de fases. Tais modelos foram baseados em modelos clássicos, que já foram validados e bastante explorados na literatura. Para o terceiro sistema dinâmico, Sistema de Tanques Interativos, foi utilizado o modelo desenvolvido e validado por Pereira (1996). Esses três modelos foram utilizados para projetar e testar os sistemas de detecção e diagnóstico de falhas propostos neste trabalho. A vantagem em se ter um modelo bem próximo da planta real, que foi uma preocupação deste trabalho, é que o projeto dos sistemas de detecção e diagnóstico de falhas podem ser feito via simulação, necessitando apenas de pequenos ajustes na implementação real.

Foi estudada uma nova estrutura de *observador de estado*, baseada na teoria de modos deslizantes, para sistemas discretos incertos, proposta por Takahashi e Peres (1997). Essa abordagem foi aplicada na estimação de estado do sistema de acionamento C.C.. Foram projetados observadores para rejeição de perturbações em uma ou duas direções dos três estados do sistema. Optou-se em utilizar a proposta de Takahashi e Peres (1997) porque, como foi provado pelos autores, a estrutura em modos deslizantes é uma generalização de várias outras encontradas na literatura. Além disso, no caso do sistema de acionamento C.C., os modelos obtidos são bastante convenientes, pois envolvem um pequeno número de operações de soma e multiplicação. Além disso, a alocação dinâmica dos autovalores do observador é feita de forma bastante simples.

Com relação à *identificação de sistemas dinâmicos*, foi apresentada uma estrutura de neurônio *fuzzy* (NFN), proposta por Yamakawa *et alii* (1992). Para esta estrutura, foi proposta uma metodologia de treinamento, onde chegou-se numa expressão para calcular a taxa de aprendizado ótima, que permite encontrar o ponto de mínimo local da função objetivo (erro quadrático médio) em apenas um passo, empregando o Método de Gradiente. Além disso, foi provada a convexidade da função objetivo, que implica que o mínimo local é mínimo global. Também foi proposta uma generalização (NFNG) da topologia original, visando a sua utilização em identificação de sistema com mais de uma saída. Finalmente, foi proposto um algoritmo bastante simples para geração das funções de pertinência, empregando uma rede neural auto-organizada. Os resultados obtidos com NFNG em identificação de sistemas dinâmicos são semelhantes aos obtidos por Narendra e Parthasarathy (1990), utilizando redes neurais multicamadas, e por Wang (1994), empregando uma rede *neurofuzzy* complexa. A vantagem do NFNG sobre estas duas estruturas é que o número de operações envolvidas no seu processamento é consideravelmente menor e exige cálculos de funções mais simples. Além disto, um outro fato que deve ser levado em consideração é que, no caso da rede multicamada e *neurofuzzy* de Wang (1994), a função objetivo, a ser minimizada no treinamento, não é convexa com relação aos pesos, implicando na existência de mínimos locais, o que dificulta a convergência para um erro exigido. Isto faz com que o treinamento do NFNG seja muitas vezes mais rápido que o treinamento da rede multicamadas tipo perceptron e a rede de Wang (1994). Por isso, o NFNG pode ser uma opção atraente em

aplicações em tempo real e em treinamento *on-line*. Os resultados apresentados não permitem fazer uma generalização quanto à capacidade de aproximação e generalização usando o NFNG, mas servem para alertar que nem sempre é preciso utilizar estruturas complexas, como as redes neurais multicamadas e *neurofuzzy* de Wang (1994), para identificar sistemas dinâmicos.

Foi proposta uma estrutura geral para detecção e diagnóstico de falhas em sistemas dinâmicos, baseada em classificação de padrões. Esta estrutura foi usada para detectar e diagnosticar falhas nos três sistemas dinâmicos modelados no capítulo 3, para três estratégias diferentes. A diferença básica de uma estratégia para outra é a forma de geração do padrão, que é o vetor de entrada do sistema de classificação. Na primeira, o padrão é formado somente utilizando as entradas e saídas da planta. Já a segunda e terceira, chamadas de estratégias híbridas, são utilizados também resíduos obtidos via observadores de estados e identificação de parâmetros, respectivamente. Cada estrutura possui vantagens e desvantagens.

A primeira estrutura apresenta as seguintes vantagens: (1) não necessitar do modelo matemático da planta, após o treinamento da rede; (2) algoritmo de treinamento bem mais simples do que aqueles usados em estratégias que utilizam redes neurais multicamadas, como é o caso dos trabalhos de Chow e Mangum (1991), Benieri *et alii* (1995) e outros, porque é um treinamento competitivo, ou seja, não utiliza derivadas; (3) geração automática do modelo qualitativo de falhas. Por outro lado, a detecção de falha, que provoca variações lentas nas grandezas envolvidas, ainda é um problema; uma idéia, a ser explorada em trabalhos futuros, é a detecção deste tipo de falha a partir da evolução das saídas da rede *neurofuzzy and/or*. Vale lembrar que tais saídas representam os graus de pertinência da operação normal e das falhas.

A segunda estrutura, baseada em observador de estado em modos deslizantes, é mais robusta a variação de parâmetros da planta e erros de modelagens que aquelas que utilizam observadores clássicos. Essa robustez também é verificada com estratégias que utilizam outras topologias de observadores robustos para entradas desconhecidas, como, por exemplo, Frank e Wünnenberg (1989) e Ge e Fang (1988). A vantagem do observador em modos deslizantes é que, no caso do sistema de acionamento C.C., o projeto fica bastante simples, podendo ser feito algebricamente. Comparando essa segunda estratégia com a primeira, a desvantagem da mesma é a necessidade do modelo matemático da planta, que tem ser discreto, caso não se queira calcular derivadas das saídas (Takahashi e Peres, 1997). Por outro lado, a aplicação dessa estratégia de detecção de falhas como ferramenta para reconfiguração do sistema de controle, para falhas em sensores, mostrou-se uma alternativa viável. Além disso, a utilização do sistema de classificação de padrão, para processar os resíduos dos observadores, permite a geração automática da lógica de decisão, que não é uma tarefa fácil (Frank, 1990). Um ponto importante a ser destacado é que a lógica de decisão obtida a partir de sistema de classificação de padrões torna o sistema de detecção mais robusto, quando comparada à utilização da lógica de detecção de falhas baseada em valores limites (limiares) para os resíduos; tal superioridade decorre da incorporação da informação do ponto de funcionamento da planta, através dos valores de entradas e saída da mesma.

A terceira estratégia é baseada em identificação do modelo da planta, obtido via treinamento do NFNG. Os coeficientes do modelo são os pesos do NFNG. As variações nesses coeficientes são usadas na geração de resíduos, que são empregados para detectar as falhas no sistema. A vantagem dessa estratégia, com relação àquelas que utilizam identificação de parâmetros empregando o método de Mínimos Quadráticos Recursivo, como, por exemplo, em Isermann (1984) e Camargo (1997), é que o número de operações utilizando o NFNG é menor; além disso, variações na planta são traduzidas de forma imediata em

variações dos pesos do NFNG. Este fato proporciona que falhas bruscas sejam detectadas de maneira quase instantânea.

A estrutura proposta apresenta ainda as seguintes vantagens:

- capacidade de aprendizado, mesmo depois que o sistema de detecção foi implementado, as falhas que ocorrerem na planta podem ser utilizadas para reajustar os pesos da rede do sistema de classificação;
- a utilização do sistema de classificação baseado na rede *neurofuzzy and/or* além de fornecer uma classificação *crisp* do funcionamento, fornece os graus de pertinência de todas as classes de saída. Em situação onde duas falhas apresentam graus de pertinência próximos, esta informação é mais confiável.

Os resultados obtidos não permitem fazer uma comparação quantitativa entre o desempenho das estratégias propostas com as diversas alternativas presentes na literatura, mas mostrou que são estruturas viáveis para serem empregadas em detecção e diagnóstico de falhas em sistemas dinâmicos. Um fato que deve ser considerado é que, apesar dos modelos utilizados serem bem próximos das plantas reais, não foram simuladas todas as situações ou perturbações que podem ocorrer na prática. Isto, eventualmente, exigiria pequenos ajustes nos pesos das redes, na fase de implementação.

Finalmente, deve-se frisar que o problema de detecção e diagnóstico de falhas em sistemas dinâmico é um tema multidisciplinar e que, apesar de existirem várias estratégias presentes na literatura, é um assunto novo, atraente, importante e que está em aberto. Os resultados obtidos com uso de sistemas de detecção de falhas têm reflexos evidentes sobre variados aspectos, que vão desde a segurança da planta até um melhor desempenho. Segundo Patton *et alii* (1989) acidentes com aviões comerciais poderiam ter sido evitados se houvesse um sistema eficiente de detecção de falhas. Vale ressaltar que esse problema não pode ser encarado como problema puramente matemático, visto que informações de especialistas do processo podem ser úteis para definição da estratégia e topologia do sistema de detecção.

Como continuidade desse trabalho propõem-se:

- comparar o desempenho das estratégias propostas com outras presentes na literatura;
- implementação prática das estratégias propostas para o sistema de tanques interativos e comparar os resultados obtidos por Pereira (1996) e Camargo (1997);
- verificar a possibilidade de se detectar falhas com evolução lenta a partir da evolução dos graus de pertinência das classes de saída do sistema de classificação *fuzzy*.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abe, S. & Lan, M. (1995). "A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification". *IEEE Fuzzy Systems*, vol. 3, No. 1, pp. 18-28.
- Aguirre, L. A.; Pereira, M. F. S. & Guimarães, R. A. (1996). "Instrument Fault Detection and Isolation in a Tank System". *XI Congresso Brasileiro de Automática*, pp. 647-652, São Paulo, Brazil.
- Anderson, J. A., *An Introduction to Neural Networks*, The MIT Press, Cambridge, Massachusetts, 1995.
- Bazaraa, M. S.; Sherali, H. D. & Shetty, C. M., *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, New York, 1993.
- Bernieri, A.; Betta, G.; Pietrosanto, A. & Sansone, C. (1995). "A Neural Network Approach to Instrument Fault Detection and Isolation". *IEEE Transactions on Instrumentation and Measurement*, vol. 44, No. 3, pp. 747-750.
- Betta, G.; D'Apuzzo, M. & Pietrosanto, A. (1995). "A Knowledge-Based Approach to Instrument Fault Detection and Isolation". *IEEE Transactions on Instrumentation and Measurement*, vol. 44, No. 6, pp. 1009-1016.
- Bose, B.K. *Power Electronic A.C. Drives*, Prentice Hall, New Jersey, 1986.
- Braga, A. R. (1994). "Implementação de Estratégias de Controle Multimalha e Multivariável". *Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte/MG, Brasil*.
- Camargo, H. A. (1997). "Detecção e Isolação de Falhas Utilizando Estimação de Parâmetros de Modelos Contínuos e Limites de Confiança Adaptativos - Aplicação a uma Planta Piloto". *Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte/MG, Brasil*.
- Caminhas, W. M.; Andrade Netto, M. L.; Costa Jr., P. P. & Tavares, H. M. F. (1994). "Redes Neurais Aplicadas ao Acionamento de Motor de Indução". *Anais do 1o. Congresso Brasileiro de Redes Neurais*, pp. 235-240, Itajubá-Minas Gerais, Brasil.
- Caminhas, W. M.; Tavares, H. M. F. & Gomide, F. A. C. (1995^a). "Competitive Learning of Fuzzy, Logical Neural Networks". *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95*, pp. 639-642, Vol. 2, São Paulo, Brazil.
- Caminhas, W. M.; Tavares, H. M. F. & Gomide, F. A. C. (1995^b). "Sistema de Classificação de Padrões Usando Redes Neuro-fuzzy e Redes Neurais". *Anais do 2o. Congresso Brasileiro de Redes Neurais*, pp. 189-194, Curitiba-Paraná, Brasil.

- Caminhas, W. M.; Tavares, H. M. F. & Gomide, F. A. C. (1996^a). "Rede Lógica Neurofuzzy: Aplicação em Diagnóstico de Falhas em Sistemas Dinâmicos". *Anais do XI Congresso Brasileiro de Automática, Vol. 1, pp. 495-464, São Paulo-São Paulo, Brasil.*
- Caminhas, W. M.; Tavares, H. M. F. & Gomide, F. A. C. (1996^b). "A Neurofuzzy Approach for Fault Diagnosis in Dynamic Systems". *Proc. IEEE International Conference on Fuzzy Systems - FUZZ_IEEE' 96, Vol. 3, pp. 2032-2037, 8-11 September, New Orleans, USA.*
- Caminhas, W. M.; Tavares, H. M. F. & Gomide, F. A. C. (1996^c). "Pattern Classification Using Neurofuzzy and Neural Networks". *Proc. Information Processing and Management of Uncertainty on Knowledge-Based Systems - IPMU'96, Vol. 1, pp. 175-180, 1-5 July, Granada, España.*
- Caminhas, W. M.; Takahashi, R. H. C.; Peres, P. L. D. & Tavares, H. M. F.; (1996^d). "Observadores de Estados do Motor de Indução com Rejeição de Perturbações Paramétricas". *Anais do XI Congresso Brasileiro de Automática, Vol. 2, pp. 1125-1130, São Paulo-São Paulo, Brasil.*
- Caminhas, W. M.; Takahashi, R. H. C.; Peres, P. L. D. & Tavares, H. M. F.; (1996^e). "Discrete-Time Sliding Modes Observers for Induction Motors". *Proc. IEEE International Conference on Control Applications, pp. 314-319, 15-18 September, Michigan, USA.*
- Caminhas, W. M.; Pereira, M. F. S.; Aguirre, L. A.; Tavares, H. M. F. & Gomide, F. A. C. (1997). "Detecção e Identificação de Falhas de Instrumentação num Sistema de Tanques Interativos Usando Rede Neurofuzzy". *Anais do 3o. Congresso Brasileiro de Redes Neurais, pp. 201-206, Curitiba-Paraná, Brasil.*
- Chang, T.; Abdel-Ghaffar, K. (1992). "A Universal Neural Net with Guaranteed Convergence to Zero System Error". *IEEE Trans. On Signal Processing, Vol 40, No. 12, pp. 3022-3031.*
- Chen, T.; Chen, H. (1995). "Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks". *IEEE Trans. On Neural Networks., Vol 6, No.4, 904-910.*
- Chow, M. & Mangum, P. M. (1991). "A Neural Network Approach to Real-Time Condition Monitoring of Induction". *IEEE Transactions on Industrial Electronics, vol. 38, No. 6, pp. 448-453.*
- Chung, F.L.; Lee, T. (1994). "A Fuzzy Learning Model for Membership Function Estimation and Pattern Classification.". *Proc. Third IEEE International Conference on Fuzzy Systems, pp. 426-431, Orlando, Florida, USA.*
- Clark, R. N. (1978^a). "A simplified Instruments Failure Detection Scheme". *IEEE Trans. Aerospace Electron. Syst., Vol. 14, pp. 558-563.*
- Clark, R. N. (1978^b). "Instrument Fault Detection". *IEEE Trans. Aerospace Electron. Syst., Vol. 14, pp. 456-465.*

- Cruz, F. R. B.; Duarte, M. A. & Costa Jr., P.P. (1990). "Simulação Digital de Correntes Descontínuas em Motores de Indução". *Anais do VIII Congresso Brasileiro de Automática*, pp. 1009-1014, Belém-Pará, Brasil.
- David, P. W. (1994). "Fuzzy Fault Tree Analysis". *Proc. 3th IEEE International Conference on Fuzzy Systems*, Orlando, Florida, USA, pp. 1899-1904.
- De Carlo, R. A.; S. H. Zak & G. P. Matthews (1988). "Variable structure control of nonlinear multivariable systems: a tutorial". *Proc. IEEE*, vol. 76, n. 3, pp. 212-232.
- Fausett, L., *Fundamentals of Neural Networks*, Prentice-Hall, 1994.
- Figueiredo, M. F. (1997). "Redes Neurais Nebulosas Aplicadas em Problemas de Modelagem e Controle Autônomo". *Tese de Doutorado, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas/SP, Brasil*.
- Figueiredo, M.; Gomide, F. C.; Pedrycz, W. (1995). "Fuzzy Neurons and Networks: Models and Learning". *Proc. Third European Congress on Intelligent Techniques and Soft Computing*, pp. 332-335, Aachen, Alemanha.
- Frank, P. M. & Keller, L. (1980). "Sensitivity Discriminating Observer Design for Instrument Failure Detection". *IEEE Trans. Aerospace Electron. Syst.*, Vol. AES-16, pp. 460-467.
- Frank, P. M. & Seliger, R. (1991) "Fault Detection and Isolation in Automatic Process". *Control and Dynamic Systems*, 49, pp. 241-287.
- Frank, P. M. (1990). "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based redundancy - A Survey and Some New Results". *Automatica*, Vol. 26, No. 3, pp. 459-474.
- Frank, P. M. & Wünnenberg, J. (1989). "Robust Fault Diagnosis Using Unknown Input Observer Schemes" *Em: Fault Diagnosis in Dynamic Systems, Theory and Applications. (R. J. Patton; P. M. Frank & R. N. Clark, Eds) - Cap. 3, pp. 47-98.*
- Ge, W.; Fang, C. Z. (1988). "Detection of Faulty Components Via Robust Observation". *Int. J. Control*, Vol. 47, pp. 581-599.
- Genther, H.; König, A., Clesner, M. (1994). "Rule Weight Generation for a Fuzzy Classification System based on Fuzzy Clustering Methods". *Proc. 3th IEEE International Conference on Fuzzy Systems*, , Orlando, Florida, USA, pp. 614-617.
- Gomide, F. A. C.; Gudwin, R. R. & Tanscheit, R. (1995). "Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações: Tutorial". *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95, pp. 1-38, Vol. 2, São Paulo, Brazil.*
- Gonzalez, A. J.; Osborne, R. L. & Lowenfeld, S. (1986). "On-Line Diagnosis of Turbine-Generators Using Artificial Intelligence". *IEEE Trans. On Energy Conversion*, Vol. EC-1, No. 2, pp. 68-74.

- Goode, P. V.; Chow, M. Y. (1994). "A Hybrid Fuzzy/Neural Systems Used to Extract Heuristic Knowledge from a Fault Detection Problem". *Proc. 3th IEEE International Conference on Fuzzy Systems*, , Orlando, Florida, USA, pp. 1731-1736
- Himmelblau, D. M. (1978). *Fault Detection and Diagnosis in Chemical and Petrochemical*, Elsevier.
- Hornik, K.; Stinchcombe, M.; White, H. (1990). "Multilayer Feedforward Networks Are Universal Approximators". *IEEE Trans. On Neural Networks*, Vol 2, pp. 359-366.
- Iserman, R. (1984). "Process Fault Detection Based on Modeling and Estimation Methods - A Survey". *Automatica*, 20, pp. 387-404.
- Ishibuchi, H.; Nozaki, K.; Yamamoto, N.; Tanaka, H. (1994). "Acquisition of Fuzzy Classification Knowledge Using Genetic Algorithms". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 1963-1968, Orlando, Florida, USA.
- Jonathan, M. V. & Ungar, L. H. (1995). "Dynamic Process Monitoring and Fault Diagnosis With Qualitative Models". *IEEE Trans. Syst. Man. Cyber.*, 25, pp. 181-189.
- Jota, F. G.; Braga, A. R. & Pena, R. T. (1995). "Performance Assessment of Advanced Process Control Algorithms Using an Interacting Tank System". *IEEE Industry Applications Society, 30th Annual Meeting*, Orlando, Florida, October 8-12.
- Keller, J.; Yager, R. & Tahani, H. (1992). "Neural Network Implementation of Fuzzy Logic". *Fuzzy Sets and Systems*, vol. 45, pp 1-12.
- Kohonen, T. (1990). "Improved Versions of Learning Vector Quantization". *International Joint Conference on Neural Networks*, San Diego, CA, pp. 545-550.
- Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice-Hall Inc., New Jersey, 1992.
- Krause, P. C., *Analysis of Electric Machinery*, Mc Graw-Hill Book Company, New York, 1986.
- Lee, C. C. (1990). "Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part I and II". *IEEE Trans. Syst. Man. and Cybernetics*, Vol. 20, No. 2, pp. 404-435.
- Leyeghi, S.; Dastmalchi, M.; Jacobs, E.; Knapp, R. B. (1994). "Pattern Recognition of the Polygraph Using Fuzzy Classification". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 1825-1829, Orlando, Florida, USA.
- Li, C. & Wu, C. (1994). "Generating Fuzzy Rules for a Neural Fuzzy". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 1719-1724, Orlando, Florida, USA.
- Lin, C. T. & Cunningham (1995). "A New Approach to Fuzzy-Neural System Modeling". *IEEE Trans. Syst. Man. and Cybernetics*, Vol. 26, No. 1, pp. 28-38.

- Lin, C. T. & Lee, C. S. G. (1991). "Neural-Network-Base Fuzzy Logic Control and Decision System". *IEEE Transactions On Computers*, Vol. 40, No. 12, pp. 1320-1336.
- Liu, Z. Qiang & Yan, F. (1997). "Fuzzy Neural Network in Case-Based Diagnostic System". *IEEE Trans. Fuzzy Systems*, vol 5, No. 2, pp. 209-222.
- Luenberger, D. G. (1971) "An introduction to observers". *IEEE Trans. Aut. Contr.*, v. 16, n. 6, pp. 596-602.
- Mateus, G. R.; Luna, H. P. L., *Programação Não Linear*, V Escola de Computação, Belo Horizonte, 1986.
- Mehra, R. K.; Peshon, I. (1971). "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems". *Automatica*, 7, pp. 637-640.
- Mihaela, U. (1994). "Diagnosis by Approximate Reasoning on Dynamic Fault Trees". *Proc. 3th IEEE International Conference on Fuzzy Systems*, , Orlando, Florida, USA, pp. 2051-2056.
- Milne, R. (1987). "Artificial Intelligence for Online Diagnosis". *IEE Proceedings*, Vol. 134, No. 4, pp. 238-244.
- Mitra, S.; Pal, S.k. (1994). "Self Organizing Neural Network As A Fuzzy Classifier". *IEEE Trans. Syst. Man. Cyber.*, Vol 24, No. 3, pp. 385-399.
- Mon, T. K. (1994). "Temporal Pattern Recognition using Fuzzy". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 432-435, Orlando, Florida, USA.
- Narendra, K. S.; Parthasarathy, K. (1990). "Identification and Control of Dynamical Systems Using Neural Networks". *IEEE Trans. On Neural Networks.*, Vol 1, No.1, pp. 4-27.
- Nozaki, K.; Ishibuchi, H. & Tanaka, H. (1994^a). "Trainable Fuzzy Classification Systems Based On Fuzzy If-Then Rules". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 498-502, Orlando, Florida, USA.
- Nozaki, K.; Ishibuchi, H. & Tanaka, H. (1994^b). "Selecting Fuzzy Rules with Forgetting in Fuzzy Classification Systems". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 618-623, Orlando, Florida, USA.
- Nunes, W. & Gomide, F. A. C. (1996). "Previsão de Temperaturas para Alimentos em Congelamento". *Relatório Técnico do Projeto Rhae-CNPq/Multibras - Processo 16081-96, DCA/FEEC/UNICAMP.*
- Patton, R. J.; Frank, P. M & Clark, R. N. (1989). *Fault Diagnosis in Dynamic Systems: Theory and Applications*, Prentice-Hall.
- Pedrycz, W. & Rocha, A. (1993). "Fuzzy- Set Based Models of Neuron and Knowledge-Based Networks". *IEEE Fuzzy Systems*, vol. 1, No. 4, 254-266.
- Pedrycz, W. (1993)., *Fuzzy Control and Fuzzy Systems*, John Wiley, & Sons, Inc., New York, USA, 1993.

- Pereira, M. F. S. (1996). "Desenvolvimento e Aplicação de um Método de Detecção e Identificação de Falhas Baseado em Observadores Dedicados". *Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte/MG, Brasil.*
- Pistauer, M. (1995). "Neural Network Design for a Process Fault-Diagnosis Module with Fuzzy System". *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95, pp. 591-594, Vol. 2, São Paulo, Brazil.*
- Polycarpou, M. M. & Vemuri, A. T. (1995). "Learning Methodology for Failure Detection and Accommodation". *IEEE Control Systems, June, pp. 16-24.*
- Silvino, J. L.; Bottura, C.P. & Resende, P. (1993) "Observadores de Fluxo e Estimação de Parâmetros da Máquina de Indução Via Modelamento Discreto Não-Estacionário". *Revista Controle & Automação, vol. 4 n. 6, pp. 11-22.*
- Simpson, P. K., *Artificial Neural Systems*, Pergamon Press, New York, 1990.
- Takahashi, R. H. C. & Peres, P. L. D. (1997) "Unknown Input Observers for Uncertain Systems: A Unifying Approach and Enhancements". *IEEE Conf. Decision and Control, Kobe-Japan, pp.*
- Tyan, C.; Wang, P. P. & Bahler, D. R. (1995). "Intelligent Control Based on Neural Fault Diagnosis and Fuzzy Fault Control". *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95, pp. 393-396, Vol.12, São Paulo, Brazil.*
- Tzafestas, S. G. (1989). "System Fault Diagnosis Using the Knowledge-Based Methodology" *In: Fault Diagnosis in Dynamic Systems, Theory and Applications. (R. J. Patton; P. M. Frank & R. N. Clark, Eds) – Cap. 15, pp. 509-572.*
- Ulieru, M. & Mrsic-Flogel, J. (1995). "Integrated Neural and Fuzzy Paradigms for Efficient Data Analysis in Diagnosis". *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95, pp. 229-232, Vol. 2, São Paulo, Brazil.*
- Ulieru, M. (1994). "Diagnosis by Approximate Reasoning on Dynamic Fuzzy Fault Trees". *Proc. Third IEEE International Conference on Fuzzy Systems, pp. 2051-2056, Orlando, Florida, USA.*
- Umano, M.; Okamoto, H.; Hatono, I. e Tamura, H. (1994). "Fuzzy Decision Trees by Fuzzy ID3 Algorithm and Its Application to Diagnosis". *Proc. Third IEEE International Conference on Fuzzy Systems, pp. 2113-2118, Orlando, Florida, USA.*
- Utkin, V. I. (1984) "Variable structure systems: present and future". *Autom. Rem. Contr., v. 44, pp. 1005-1020.*
- Vescovi, M. R. & Robles, J. P. (1992). "FTD: A fuzzy Diagnosis Tool". *Anais do IX Congresso Brasileiro de Automática, pp. 834-839, Vitória – Espírito Santo, Brasil.*

- Vinson, J. M. & Ungar, L. H. (1995). "Dynamic Process Monitoring and Fault Diagnosis With Qualitative Models". *IEEE Trans. Syst. Man. and Cybernetics*, Vol. 25, No. 1, pp. 181-189.
- Wang, L. X. ; Mendel, J. M. (1992). "Generating Fuzzy Rules by Learning from Examples". *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22, No. 6, pp. 1414-142.
- Wang, L. X., *Adaptive Fuzzy Systems and Control*, Prentice-Hall, 1994.
- Weber, D. P. (1994). "Fuzzy Fault Tree". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 1899-1904, Orlando, Florida, USA.
- Willsky, A. S. (1976). "A Survey of design Methods for Failure Detection in Dynamic Systems". *Automatica*, 12, pp. 601-611.
- Wu, Xing & Wu, C. (1994). "A Fuzzy Chip-Based Real-Time Fault Classifier in a Power Controller". *Proc. Third IEEE International Conference on Fuzzy Systems*, pp. 1520-1525, Orlando, Florida, USA.
- Yamakawa, T.; Uchino, E.; Miki, T. & Kusanagi, (1992). "A Neo Fuzzy Neuron and its Applications to System Identification and Predictions to System Behavior". *Proc. Of the 2nd IZUKA, Iizuka-Japan* , pp. 477-483.
- Zadeh, L. (1973). "Fuzzy Sets – Information and Control". *IEEE Trans. Syst. Man. and Cybernetics*, Vol. SMC-3, pp. 28-44.
- Zhang, C. Q.; Fadali, M. S. (1996). "Nonlinear System Identification Using a Gabor/Hopfield Network". *IEEE Trans. Syst. Man. Cyber.*, Vol 26, No. 1, pp. 124-134.
- Zhang, C. Q.; Fadali, M. S. (1996). "Nonlinear System Identification Using a Gabor/Hopfield Network". *IEEE Trans. Syst. Man. Cyber.*, Vol 26, No. 1, pp. 124-134.
- Zuben, F. V. (1993). "Redes Neurais Aplicadas ao Controle de Máquinas de Indução". *Dissertação de Mestrado, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas/SP, Brasil.*

APÊNDICE A
PARÂMETROS DOS SISTEMAS DINÂMICOS

A1 – PARÂMETROS DO SISTEMA DE ACIONAMENTO C.C.

<i>PARÂMETROS DA MÁQUINA</i>	<i>VALOR</i>
Resistência do circuito da armadura - r_a	$4.67 \times 10^{-3} \Omega$
Resistência do circuito de campo - r_{fd}	12.1875 Ω
Indutância do circuito da armadura - L_a	$2.38 \times 10^{-4} \text{H}$
Indutância do circuito de campo - L_{fd}	8.750 H
Indutância mútua armadura/campo - L_{afd}	0.23 H
Momento de inércia do sistema motor/carga - J_m	2.580 Kg.m^2
Tensão nominal de alimentação do circuito da armadura - V_{an}	750 V
Tensão nominal de alimentação do circuito de campo - V_{fdn}	750 V
Corrente nominal de alimentação do circuito da armadura - I_{an}	17 098 A
Corrente nominal de alimentação do circuito de campo - I_{fdn}	61.53 A

<i>PARÂMETROS DO CONTROLADOR DE VELOCIDADE</i>	<i>VALOR</i>
K_p	10 000
T_i	1
Faixa de variação da saída do controlador (I_a^*)	$[-3 I_{an} \text{ a } 3 I_{an}]$
Período de amostragem	2 ms

<i>PARÂMETROS DO CONTROLADOR DE CORRENTE DE ARMADURA</i>	<i>VALOR</i>
K_p	0.1
T_i	0.02
Faixa de variação da saída do controlador (α_a)	$[0 \text{ a } \pi]$
Período de amostragem	2 ms

<i>PARÂMETROS DO CONTROLADOR DE CORRENTE DE CAMPO</i>	<i>VALOR</i>
K_p	100
T_i	1
Faixa de variação da saída do controlador (α_{fd})	$[0 \text{ a } \pi]$
Período de amostragem	2 ms

A2 – PARÂMETROS DO SISTEMA DE ACIONAMENTO C.A.

PARÂMETROS DA MÁQUINA	VALOR
Resistência do estator - R_s	1.5 Ω
Resistência do rotor - R_r	1.6 Ω
Indutância própria do estator - L_{ss}	0.115 H
Indutância própria do rotor - L_{rr}	0.116 H
Indutância mútua entre o estator e rotor - M	0.098 H
Número de pólos da máquina - P :	4
Momento de inércia - J_m	0.008 Kg.m ²
Tensão de estator nominal v_n	220 V
Frequência nominal	60 Hz

A3 – PARÂMETROS DO SISTEMA DE TANQUES INTERATIVOS

PARÂMETROS DO SISTEMA DE TANQUES INTERATIVOS	VALOR
Vazão máxima de entrada do tanque TQ-2 - Q_{imax}	1.06 x 10 ⁻³ m ³ /s;
Vazão máxima de saída do tanque TQ - Q_{omax}	1.3 x 10 ⁻³ m ³ /s
Área da base dos tanques TQ-2 e TQ-3 - A	0.43 m ²
Alcance da válvulas FCV-1 - R_1	124
Alcance da válvulas FCV - R_2	11
Razão entre a queda de pressão mínima e máxima da válvula FCV-1 - p_1	0.066
Razão entre a queda de pressão mínima e máxima da válvula FCV-2 - p_2	0.52
Resistência hidráulica do duto entre os tanques - R_h	270

PARÂMETROS DO CONTROLADOR DE NÍVEL DE TQ-3	VALOR
Kp	8
Ti	130
Faixa de variação da saída do controlador (u_i)	[0 a 1]
Período de amostragem	2 segundos

PARÂMETROS DO CONTROLADOR DE VAZÃO DE SAÍDA DE TQ-3	VALOR
K_p	0.4
T_i	3
Faixa de variação da saída do controlador (u_2)	[0 a 1]
Período de amostragem	15 segundos

- PARÂMETROS INDICADORES DE FALHAS

TIPO DE FALHA	PARÂMETRO	DEFINIÇÃO DO PARÂMETRO
agarramento da válvula CV-1	q_{a1}	[0.09 a 0.45]
agarramento da válvula CV-2	q_{a2}	[0.415 a 0.472]
desligamento da bomba BA-01	k_{ba1}	{0,1}
desligamento da bomba BA-02	k_{ba2}	{0,1}
entupimento no duto entre TQ-1 e TQ-2	k_{fe12}	[0.10 a 0.87]
entupimento no duto entre TQ-3 e TQ-1	k_{fe31}	[0.11 a 1.00]
entupimento no duto entre TQ-2 e TQ-3	k_{fe23}	[0.70 a 0.95]
vazamento no duto entre TQ-2 e TQ-3	k_{fv23}	[0.20 a 1.00]
falha no sensor de nível de TQ-3	k_{fy1}	{0,1}
falha no sensor de vazão de saída de TQ-3	k_{fy2}	{0,1}
falha no sensor de vazão de entrada de TQ-2	k_{fy3}	{0,1}

APÊNDICE B
PARÂMETROS DO NFNG PARA O EXEMPLO 1

O sistema dinâmico do exemplo 1, apresentado no capítulo 5, é definido pela equação a diferenças:

$$y(k+1) = \alpha_1 \cdot y(k) + \alpha_2 \cdot y(k-1) + g[u(k)] \quad (\text{B.1})$$

sendo

$$g(u) = 0.6 \text{sen}(\pi u) + 0.3 \text{sen}(3\pi u) + 0.1 \text{sen}(5\pi u) \quad (\text{B.2})$$

O modelo identificado possui a seguinte expressão:

$$\hat{y}(k+1) = \alpha_1 \cdot \hat{y}(k) + \alpha_2 \cdot \hat{y}(k-1) + \hat{g}[u(k)] \quad (\text{B.3})$$

sendo $\hat{g}[u(k)]$ a função identificada pelo NFNG. Para os resultados apresentados no capítulo 5, o número de partições *fuzzy* da entrada foi igual a vinte.

Fazendo $x_1 = u(k)$, tem-se a seguinte expressão para $\hat{g}[u(k)] = \hat{g}(x_1)$.

$$\hat{g}[u(k)] = \hat{g}(x_1) = f_1(x_1) = \mu_{k_1}(x_1) \cdot w_{1k_1} + \mu_{k_1+1}(x_1) \cdot w_{1k_1+1} \quad (\text{B.4})$$

onde

$$\mu_{1j}(x_1) = \begin{cases} \alpha_{21j} \cdot (x_1 - a_{1j}) + 1, & a_{1j-1} \leq x_1 \leq a_{1j} \\ \alpha_{1j} \cdot (x_1 - a_{1j}) + 1, & a_{1j} < x_1 \leq a_{1j+1} \\ 0 & \text{caso contrário} \end{cases} \quad (\text{B.5})$$

e w_{1k_1} e w_{1k_1+1} são os pesos da primeira e segunda partição ativa, respectivamente.

Os parâmetros $(w_{1j}, a_{1j}, \alpha_{e1j}$ e $\alpha_{d1j})$ obtidos com o treinamento estão presente na tabela B.1.

Tabela B1: Parâmetros do NFNG

w_{1j}	a_{1j}	α_{e1j}	α_{d1j}
-0.0055	-1.000	9.5	-9.5
-0.5708	-0.8947	9.5	-9.5
-0.4433	-0.7895	9.5	-9.5
-0.6618	-0.6842	9.5	-9.5
-0.3935	-0.5789	9.5	-9.5
-0.4004	-0.4737	9.5	-9.5
-0.3927	-0.3684	9.5	-9.5
-0.5372	-0.2632	9.5	-9.5
-0.7124	-0.1579	9.5	-9.5
-0.3514	-0.0526	9.5	-9.5
0.3247	0.0526	9.5	-9.5
0.7090	0.1579	9.5	-9.5
0.5480	0.2632	9.5	-9.5
0.3855	0.3684	9.5	-9.5
0.3985	0.4737	9.5	-9.5
0.3952	0.5789	9.5	-9.5
0.4261	0.6842	9.5	-9.5
0.6363	0.7895	9.5	-9.5
0.6155	0.8947	9.5	-9.5
0.0466	1.000	9.5	-9.5